4D v11 SQL

Design Reference
Windows®/Mac OS®

4D®

4D v11 SQL - Design Reference
For Windows® and Mac OS®

IMPORTANT LICENSE INFORMATION

Use of this software is subject to its license agreement included with the software. Please read the License Agreement carefully before using the software.

Contents

Preface

4D is a powerful relational database application, a high-performance SQL engine, a complete multi-platform development tool and a dynamic Web server.

You can use 4D to manage your own data or develop custom applications for different kinds of database management tasks.

For example, you can:

- Create a database structure of tables and fields,
- Design forms for entering, modifying, and displaying records,
- Search and sort records,
- Create reports and labels from information in the database,
- Import and export data between 4D databases and other applications,
- Publish your database on the World Wide Web.

With 4D, you can enhance conventional data management tasks with the following features:

- The powerful Form Wizard that lets you create sophisticated forms and reports with only point-and-click operations,
- A password access system to protect sensitive data,
- An integrated environment for generating charts that lets you produce a number of different chart types, extracted from the data,
- The capability to create custom applications with your own menus, dialog boxes, toolbars, and buttons; these applications can be compiled and distributed to other users,

- The possibility of using innumerable *Web Services* published on the Web within your database,
- A full-featured programming language that lets you incorporate commands or functions written in other languages.
- An SQL engine in compliance with standards which permits the creation of tables and fields as well as the handling and query of the database data.
- A centralized application verification and administration window.

Novice users can quickly create databases and begin managing their data. Experienced users can customize their databases with 4D's development tools. More experienced developers can use 4D's powerful programming language or SQL to add sophisticated features and capabilities to their databases, including file transfer, communications, and World Wide Web capabilities.

When you create a custom database, you can use custom menus and dialog boxes, button palettes, toolbars, and multiple windows to enhance your databases and make users more productive.

About the Manuals

The manuals described below provide a guide to the features of both 4D and 4D Server. The only exception is the *4D Server Reference* which describes features exclusive to 4D Server.

- The *Self-training* manual leads you step-by-step through various exercises that will allow you to familiarize yourself quickly with the functionalities and basic concepts of 4D.
- The *Design Reference* manual describes the Design mode and all the operations that you can carry out in this environment.
- The *4D Language Reference* manual is the guide to 4D language. Use this manual to learn how to customize a database by incorporating 4D commands and functions.
- The *4D SQL Reference* manual presents the various ways of querying the integrated SQL engine of 4D and provides the syntax of the SQL commands and functions that are supported. It also includes a tutorial that allows you to familiarize yourself with the use of this language in 4D.

- The *XML Keys-BuildApplication* manual lists and describes the XML keys that can be used to configure the application builder of 4D.
- The *XML Keys-Backup* manual lists and describes the XML keys that can be used to set the 4D backup preferences.
- The *4D Server Reference* is a guide to installing 4D Server and managing multi-user databases with 4D Server.
- Moreover, additional manuals describing the functioning of the plug-ins and utility programs delivered with 4D and 4D Server (*4D Chart Language, 4D Pack*, etc.) are available.

About this Manual

This manual describes the 4D Design environment. The Design environment is where you set up the structure of your application, create its interface and program the mechanisms and processes that it will handle. You can also test its functioning at any time and use various integrated tools for data entry, display and graphic representation of records.
Once the development of your application is finished, you can execute it in *Application mode*, an environment where only the interface elements and functions that you have chosen to display will be accessible to the user.

Cross-Platform

This manual explains the use of 4D on both the Mac OS and Windows platforms. Although the concepts and functionality of both versions of 4D are nearly identical, the manual addresses any differences where necessary. Such differences include the graphical user interface and keyboard commands.

This manual contains graphics illustrating the Windows environment. If there is a substantial difference in the appearance of the Mac OS version of a window or dialog box, it is also presented.

Right-click and Control+click under Mac OS

Certain functions are activated (for instance, the display of contextual menus) by carrying out a **right-click**, i.e. a click with the right button of the mouse, under both Windows and Mac OS.
Under Mac OS, the same result can be obtained by using a **Control+click** combination. This alternative is necessary when you are using a single-button mouse.

Chapter Descriptions

This manual is divided into the following chapters:

- **Chapter 1, "Managing 4D Files,"** introduces basic 4D operations such as starting the application and handling disk files.
- **Chapter 2, "Overview of Design Environment,"** introduces the interface and editors as well as search and navigation tools in the Design environment.
- **Chapter 3, "Setting Preferences,"** describes each modifiable parameter in the Preferences of 4D.
- **Chapter 4, "Database Structure,"** introduces the Structure editor and explains how to create tables, fields, and related tables.
- **Chapter 5, "Creating Forms,"** introduces the Form Wizard and explains how to create forms.
- **Chapter 6, "Form Editor,"** explains how to use the Form editor to set form properties, create and modify form objects, and set object properties; it also describes the creation and use of object libraries.
- **Chapter 7, "Fields and Active Objects in Forms,"** explains how to set and enforce business rules for fields and enterable objects and how to add custom interface elements such as combo boxes, drop-down lists, hierarchical lists, tab controls, and picture buttons.
- **Chapter 8, "Output Displays and Reports,"** explains how to create a form for printing a report. It includes an explanation of how to create subtotals and other summary calculations using methods.
- **Chapter 9, "User Forms,"** describes the editor that lets users customize application forms.
- **Chapter 10, "Quick Reports,"** describes how to work with the integrated Quick report editor.
- **Chapter 11, "Methods,"** introduces the 4D Method editor and explains how to use it to create methods.
- **Chapter 12, "Users and Groups,"** explains how to use the password access system to create a system that controls access to tables, table operations, forms, methods, menu commands, and plug-ins.
- **Chapter 13, "Custom Menus,"** explains how to use the Menu Bar editor to create custom menus. It also explains how to use connected menus to simplify menu management.

- **Chapter 14, "Picture Library,"** explains how to create and manage pictures for use as buttons, icons, and background pictures.
- **Chapter 15, "Lists,"** explains how to use the Lists editor to create choice lists to simplify data entry.
- **Chapter 16, "Style Sheets,"** explains how to use the Style sheets editor to set up interfaces adapted to each platform.
- **Chapter 17, "Processes,"** introduces the concept of multi-tasking in 4D using processes. It explains how to start a process and how to use the Runtime Explorer to view process information and control process execution.
- **Chapter 18, "Record Management,"** introduces the standard interface of 4D which can be used for selecting, sorting, querying, adding and deleting records.
- **Chapter 19, "Importing and Exporting Data,"** describes the importing and exporting of records using the 4D editors.
- **Chapter 20, "Labels,"** describes how to use the Label editor to create and print labels.
- **Chapter 21, "Graphs,"** explains how to represent the information of your database graphically using the integrated4D Chart plug-in.
- **Chapter 22, "Publishing or Using Web Services,"** explains the advantages of Web Services and describes how to publish methods as Web Services or call external Web Services from within your databases.
- **Chapter 23, "Administration of Integrated Servers,"** describes the integrated administration commands of the 4D Web server and 4D SQL server.
- **Chapter 24, "Backup and Restoring of the Database,"** introduces the commands for launching and handling database backups as well as the mechanism for restoring archives in case of an incident.
- **Chapter 25, "Maintenance and Security Center,"** details each page of the MSC window.
- **Chapter 26, "Compilation,"** details the integrated functions that can be used to compile your database.
- **Chapter 27, "Finalizing and Deployment of 4D Applications,"** explains the operation of the 4D application builder.

- **Chapter 28, "4D Components,"** describes the principles of designing, creating and installing 4D components.
- **Appendix A, "Assigning a Custom Help File,"** explains how to install a custom help file in a 4D database.
- **Appendix B, "Command Line Interface (CLI) Under Mac OS,"** provides information about and descriptions of 4D-related Mac OS X Terminal command lines.
- **Appendix C, "XLIFF Architecture,"** describes the XLIFF standard and explains how it is used in 4D.
- **Appendix D, "Use of Specialized Dictionaries,"** provides information about the installation, loading and use of specialized dictionaries.

Conventions

All the manuals in your documentation package, including this one, use certain conventions to help you understand the material.

The following explanatory notes are used:

Note Text emphasized like this provides annotations and shortcuts that will help you use 4D more productively.

4D Server Throughout the manual, 4D and 4D Server/4D Client are referred to simply as 4D. Differences between the operation of the two products are explained in 4D Server notes which provide information about using 4D Server/4D Client. This information is provided only when the operation of 4D Server/4D Client differs from that of 4D.

Warnings like this alert you to situations where data might be lost.

In addition, names of tables in a database are shown in brackets in the text to help distinguish them from the names of fields, forms, and other items. For instance, the Company table is written as the [Company] table.

1 Managing 4D Files

This chapter presents the execution principles and general architecture of 4D applications. It also describes the installation of plug-ins into this architecture as well as the process for converting databases created using previous versions of 4D. The following subjects will be covered:

- Starting 4D,
- Creating a database,
- Opening an existing database,
- Description of 4D database files,
- Changing the data file,
- Installing plug-ins,
- Converting databases created in previous versions.

Unless otherwise specified, all the instructions and explanations are applicable both to 4D and 4D Server.

Starting 4D

Once you have installed and registered your version of 4D as described in the installation manual, you can start the application.

To do so, you can select the application icon and choose **Open** in the **File** menu of the operating system, or double-click directly on this icon.

By default, 4D displays a dialog box that you can use to create a new database using a predefined template. To create a database, choose a template and click on **Create database**. You must then indicate the name and location of the database. For more information about using database templates, please refer to the paragraph "Creating a Database from a Template" on page 34.
You can choose to no longer display this window by checking the "Do not display this dialog box on startup". In this case, on startup 4D will display the Open file dialog box directly, or the option chosen in the Preferences (see the paragraph "Options Page" on page 163).

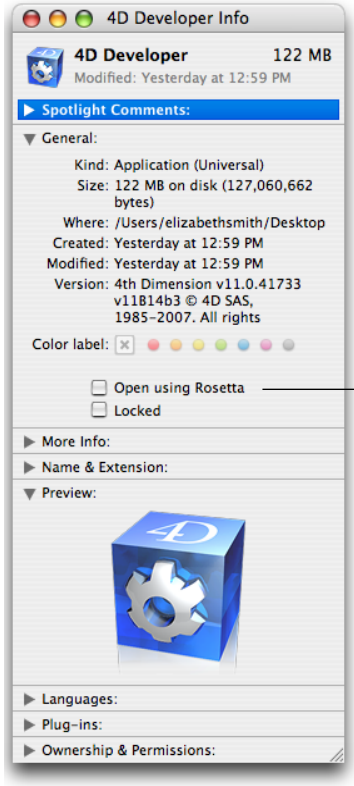
If you click on **Cancel** (or if you choose the "Do nothing" option in the Preferences), the 4D application is launched but no database is opened. In this case, you can open a database via the **Maintenance and security center** (see the paragraph "Display in Maintenance Mode" on page 1210) or create/open a database using the commands of the **File** menu or the tool bar. For more information about creating and opening a database, please refer to the following paragraphs.

Note You can configure the functioning of 4D on startup via the Preferences. For more information, please refer to the paragraph "Options Page" on page 163.

Universal Binary Architecture (Mac OS)

The Universal Binary architecture of the Mac OS version of 4D v11 means that 4D applications run in native mode on recent Apple machines based on Intel processors ("Mac Intel"). This results in a considerable gain in performance.

By default, 4D v11 applications are executed in native mode in this environment. However, you can force execution in emulated mode (Rosetta mode), more particularly in order to use plug-ins not having Universal Binary architecture. To do this, select the 4D application on the desktop and choose the **Get Info** command in the **File** menu. An option in the "Info" window can be used to force the application to run in emulated mode:



Execution in emulated mode (Rosetta mode)

Creating a Database

You can create a database using a template already containing elements (tables, fields, etc.) that you can simply adapt to your own needs. You can also create a blank database. Finally, you can create a database from a structure definition that has previously been exported.

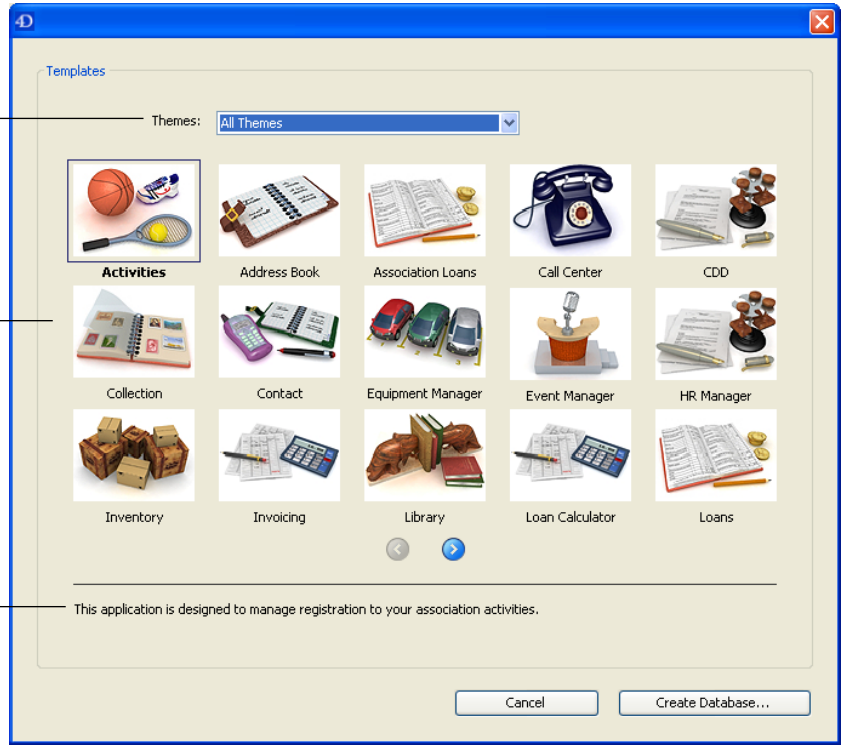
Creating a Database from a Template

The 4D database templates are directly operational. Each template includes an interface, tables, fields, forms, methods, etc., that can serve as prototypes for custom developments. The possibility of creating databases from templates also allows novice users to see the potentialities of 4D.

► To create a database from a template:

1 **Choose New > Database From Template... in the File menu or in the menu associated with the "New" button of the 4D tool bar.**

The dialog box for choosing the template appears:

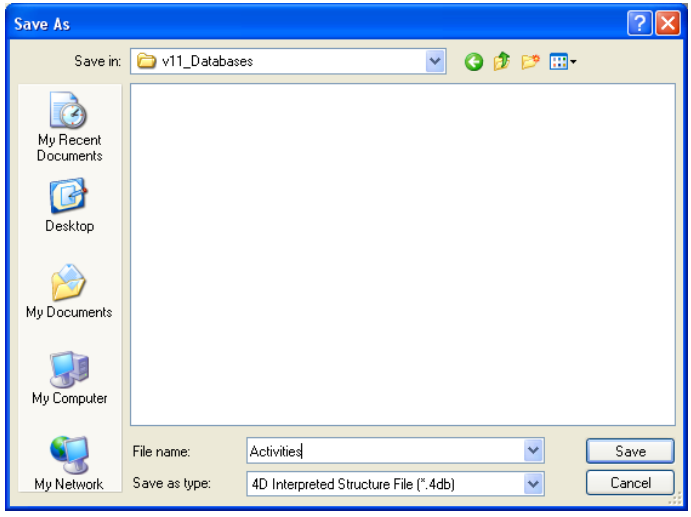


Notes

- By default, this dialog box appears on each opening of the application (see the paragraph “Starting 4D” on page 31). It is possible to disable this functioning.
- Database templates are stored in a folder named “4D Modèles” or “4D Templates”. This folder must be placed at the same level as the .exe file of the 4D application (under Windows) or the package (under Mac OS). If this folder is missing, an empty window is opened.

2 Select a template from the themes provided and click on Create Database...

A standard Save As dialog box appears so that you can choose the name and location of the 4D database:



3 Enter the name of your database and click Save.

You can choose any file name that is accepted by your operating system.

When you validate the dialog box, 4D closes the current database (if any), creates a folder at the location indicated (the name of the folder is the database name) and puts all the files needed for proper operation of the database into it. For more information about the architecture of 4D databases, please refer to paragraph “Database Architecture” on page 42.

The splash screen of the Application mode of the database then appears. You can begin using the database, switch to Design mode (via the **Mode** menu), add fields, tables, etc.

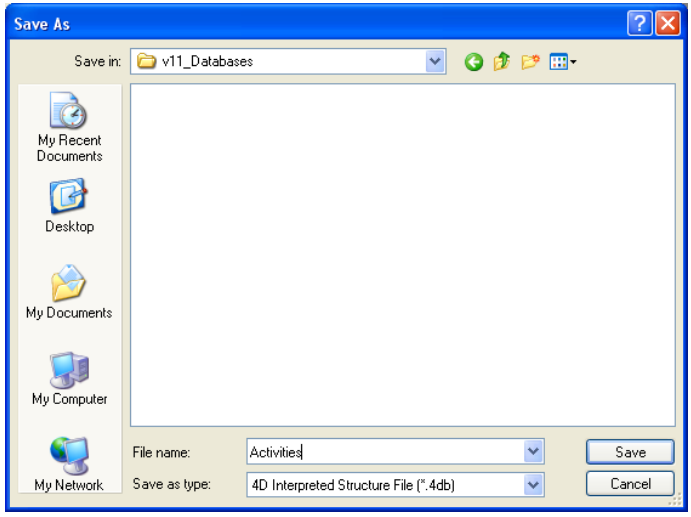
Creating a Blank Database

You can create a blank database, i.e. that does not contain any tables, fields or forms. Its the equivalent of a blank sheet for databases.

► To create a blank database:

1 **Choose New > Database... from the File menu or from the menu associated with the "New" button in the 4D tool bar.**
You can also click directly on the "New" button of the tool bar.

A standard Save as dialog box appears so that you can choose the name and location of the 4D database:



2 **Enter the name of your database and click Save.**

You can choose any file name that is accepted by your operating system.

When you validate the dialog box, 4D closes the current database (if any), creates a folder at the location indicated (the name of the folder is the database name) and puts all the files needed for proper operation of the database into it. For more information about the architecture of 4D databases, please refer to the paragraph "Database Architecture" on page 42.

The 4D application window is then displayed with the Explorer in the foreground. You can then, for example, create project forms or display the Structure editor and add tables, fields, etc.

Creating a Database from a Structure Definition

Structure definitions that are exported in XML format can be used to create new identical databases on the fly. In this case, the structure definition can be considered as a structure template, which it is possible to duplicate at leisure.

Note For more information about exporting and importing structure definitions, please refer to the paragraph "Exporting and Importing Structure Definitions" on page 330.

An XML structure definition can be used as such or can be modified beforehand via an XML editor. This means that the use of any type of mechanism used to generate structures by programming can be considered.

Furthermore, since the internal format of 4D structure definition XML files is public (see the paragraph "Format of a 4D Structure Definition" on page 331), it is possible to build this type of file from other database environments or any design application in order to generate 4D databases automatically.

► To create a database from a structure definition:

1 **Select New > Database From Structure Definition... from the File menu of 4D or from the menu associated with the "New" button in the 4D tool bar.**

A standard Open document dialog box appears so that you can specify the definition file to be opened. You must select an XML format file that respects the "grammar" of 4D structure definitions (the program validates the file via the DTD).

2 **Select a structure definition XML file then click OK.**

4D displays a dialog box that can be used to choose the name and location of the database to be created.

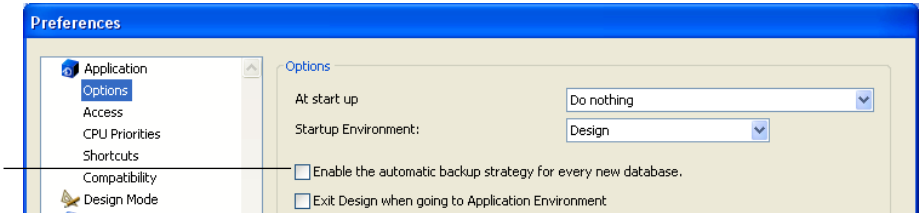
3 **Choose the name and location of the database to be created then click on Save.**

If the XML file is valid, 4D closes the current database (if any) and creates a new structure based on the structure definition. A folder is created at the location indicated (the name of the folder is the database name) that contains all the files needed for proper operation of the new database. The Explorer window is then displayed.

Automatic Backup Strategy

By default, 4D enables a default backup strategy when the database is created. This strategy implements a set of backup parameters ensuring the minimum security of the database.

It is possible to modify this functioning by unchecking the **Enable the automatic backup strategy for every new database** option of the application Preferences (Application/Option page):



The default backup strategy includes the following parameters:

- Creation and use of a log file (named *DataFileName.journal*).
- Backup of all the database files (data, log, structure and user structure) next to the structure file of the database.
- Automatic backup each Sunday at midnight (00:00:00).
- Default backup parameters (storing of 3 sets of backups, average rates, etc.).
- All automatic restore options.

Naturally, it is possible to modify these parameters after creation of the database, in the “Backup” theme of the database Preferences.
For more information about the integrated backup module of 4D, please refer to the chapter “Backup and Restoring of the Database” on page 1169.

Opening an Existing Database

There are several possibilities for opening existing databases. You can use the standard Open dialog box or functions that allow you to open recently used databases directly. Note that you can set the default functioning of the 4D application on startup (see the paragraph “Options Page” on page 163).

4D Server For more information about opening databases with 4D Client, please refer to the *4D Server Reference* manual.

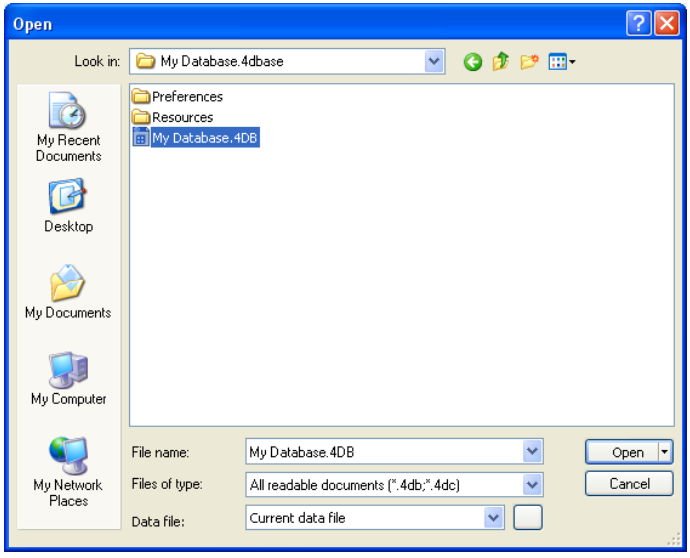
Opening a Database via the Open Dialog Box

The standard Open dialog box can be used to select any 4D database and provides several opening options.

► To open an existing database using the Open dialog box:

1 **Select Open > Database... in the File menu or 4D or Database... in the menu associated with the "Open" button of the 4D tool bar.**

The following dialog box appears:



2 **Select the database to open.**

- Under Windows, you must select the database structure file (suffixed .4DB or .4DC).
- Under Mac OS, you can select either the database package (suffixed .4dbase), or the database structure file (suffixed .4DB or .4DC). For more information about package architecture, please refer to the paragraph ".4dbase Extension" on page 45.

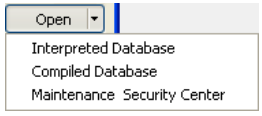
3 **Choose an opening option (see the next paragraph) or click on Open to open the database.**

Note You can use drag and drop functions or double-click on a .4dbase package or a .4DB or .4DC structure file in order to launch a 4D application without passing through the opening dialog box.

Open Dialog Box Options

In addition to the standard system options, the Open dialog box of 4D provides two menus of specific options that can be accessed via the **Open** button and the **Data file** menu.

- **Open**: The menu associated with this button can be used to specify the opening mode of the database. It includes the **Interpreted Database** and/or **Compiled Database** options as well as the **Maintenance Security Center**:



  - **Interpreted Database** and **Compiled Database**: These two options are provided when the selected database contains both interpreted and compiled code. If one of these modes is not available (database not compiled or not recompiled, compiled file only, etc.), the corresponding option will not appear.

Note 4D also lets you switch from interpreted to compiled mode (and back) during use. For more information, please refer to the paragraph "The Run Menu" on page 90.

  - **Maintenance Security Center**: Opening in secure mode allowing access to damaged databases in order to carry out any necessary repairs. For more information, please refer to the paragraph "Display in Maintenance Mode" on page 1210.

When you click directly on the **Open** button, the database is opened in the last opening mode used.

- **Data file**: This menu can be used to specify the data file to be used with the database. By default, the **Current data file** option is selected. The database opens with this file is you click on **Open**.
This menu includes two additional options since 4D allows you to use the same structure file with different data files.
  - **Choose another data file**: This option can be used to open the database with an existing data file other than the current one. When you select this option and then click on **Open**, the standard Open data file dialog box appears so that you can designate a data file.
  - **Create a new data file**: This option lets you create a blank data file for the database. When you select this option and then click on **Open**, the standard Save data file dialog box appears so that you can name and indicate the location of the data file to be created.

Once you have changed the current data file, 4D will open it by default subsequently, unless it finds a file having the same name as the structure file followed by ".4DD" located in the same directory as the structure file. If you move or rename the data file, you will need to locate it again.

Note It is possible to change the data file using a keyboard shortcut on database startup. For more information, please refer to the paragraph "Changing the Data File" on page 46.

You can view the current data file of the database at any time on the "Information/Program" page of the Maintenance and security center (see the paragraph "Program and Tables" on page 1212).

Database Opening Shortcuts

4D provides several functions that can be used to open databases directly, without passing through the Open dialog box: drag and drop, the "Open Recent Databases" menu and the automatic opening of the last database used.

► To open a database using drag and drop:

1 Select the database file to be opened and drop it onto the 4D application icon.

- Under Windows, you must select the database structure file (suffixed .4DB or .4DC).
- Under Mac OS, you can select either the database package (suffixed .4dbase), or the database structure file (suffixed .4DB or .4DC). For more information about package architecture, please refer to the paragraph ".4dbase Extension" on page 45.

► To open a recently used database:

1 Select Open Recent Databases > *Database Name* in the File menu of 4D or in the menu associated with the "Open" button of the 4D tool bar.

The first time the application is launched, this menu is empty. It will be filled in gradually as different databases are used; 4D automatically memorizes the names of all the databases that are opened.

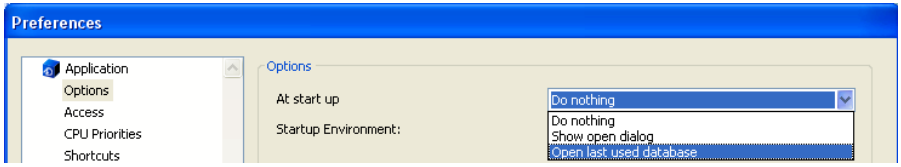
Select the **Clear Menu** command (only available via the **File** menu) in order to remove the database names from the menu.

► To automatically open the last database used:

1 **Display the "Application/Options" page of the application Preferences.**

For more information about displaying the Preferences, please refer to the paragraph "Access to the Preferences" on page 161.

2 **In the "Options" area, choose Open last used database in the "At start up" menu:**



4D will then automatically open the last database used each time the application is launched.

Note To force the display of the Open dialog box when this option has been selected, hold down the **Alt** (Windows) or **Option** (Mac OS) key during opening of the application.

4D Database Files

4D creates several files and folders by default for each database. Additional files are also used when the database has been converted from a previous version.

Database Architecture

When you create a database, the following files and folders are generated by default on the disk:

■ A **structure file**(".4DB" extension)
The structure file contains all the specifications concerning the database structure (tables, fields, field properties), forms, methods, menus, passwords and choice lists. The structure file takes the name that you enter in the database creation dialog box, followed by ".4db".

- A **data file** (".4DD" extension)
  The data file contains the data that are entered in the records and all the data that belong to the records. The data file takes the name that you entered in the database creation dialog box, followed by ".4db". When you open a 4D structure file, the application opens the current data file by default. If you change the name or location of this file, the Open data file dialog box will then be displayed so that you can select the data file that you want to use or create a new one (see the paragraph "Changing the Data File" on page 46).
  It is possible associate a structure file strictly with one data file in order to avoid any incorrect handling (see the paragraph "Associating a Data File with a Structure" on page 47).

- A **structure index file** ("**.4DIndy**" extension) and a **data index file** ("**.4DIndx**" extension)
  Indexes created in a 4D database are stored as separate files. These files are automatically placed next to the structure file. They must not be moved or renamed; otherwise, 4D will have to create them again.
  ***DatabaseName*.4DIndx** contains the data indexes and ***DatabaseName*.4DIndy** contains the structure indexes (used in particular when searching the structure).
  One of the main advantages is that if the index becomes corrupted, it is possible to physically remove the files before launching 4D so that it will be automatically created anew.

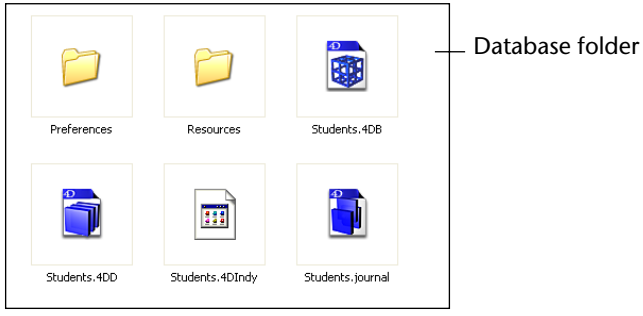
- A **data log file** (".journal" extension) — file created only when the database uses a log file.
  The log file is used to ensure the security of the database data between two backups. All the operations carried out on the data of the database are recorded sequentially in this file. So each operation on the data causes two simultaneous actions: the first on the database data (the statement is executed normally) and the second in the log file (a description of the operation is recorded). The log file is constructed independently, without disturbing or slowing down the user's work. A database can only work with one log file at a time.
  The log file records operations such as additions, modifications or deletions of records, transactions, etc.
  The log file is generated by default when a database is created. For more information, please refer to the paragraph "Managing the Log File" on page 1188.

- A **Resources folder**, containing external resources.
  The **Resources** folder, which must be located next to the database structure file (.4db or .4dc), is intended to contain all the files needed for the database interface.
  In this folder, you can place all the files needed for the translation or customization of the application interface (picture files, text files, XLIFF files, etc.). 4D uses automatic mechanisms to make use of the contents of this folder, in particular for the handling of XLIFF files (see Appendix C “XLIFF Architecture” on page 1293) and pictures (see the paragraph “Automatic Referencing of Picture Files” on page 456).

- A **Preferences folder**, containing the configuration files specific to the database — folder only created when necessary.
  The Preferences folder stores the configuration files of the database, generally in XML format. It is created automatically, more specifically when using the backup function (*Backup.xml* file) or the application builder function (*BuildApp.xml* file).

  These files are initially placed in the same folder when the database is created; it is recommended to not move them so as not to affect the automatic mechanisms of 4D.

Example of files and folders created by default for a new database



Additional Files (Converted Databases)

When you use a database created with a previous version of 4D, additional files are present:

- A **structure resources file** (“.RSR” extension)
  This file contains the “former” Macintosh type resources associated with the database structure.
  The .4DB and .RSR files must always be placed in the same directory and have the same name, otherwise you will not be able to open the database.

- A **data resources file** (".4DR" extension)
  This file contains the "former" resources associated with the data of the database.

Note For more information about converting databases, please refer to the paragraph "Converting Databases from Previous Versions" on page 51.

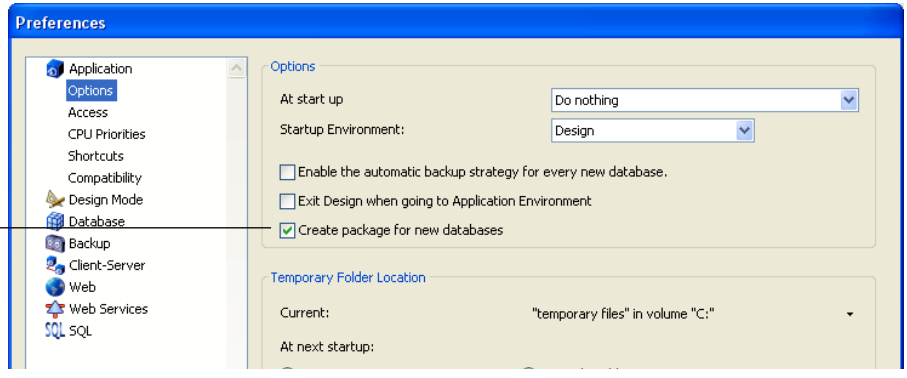
.4dbase Extension

By default, 4D databases are automatically created in a folder suffixed **.4dbase**. For example, a database named "Invoices" will be created in the folder [*Invoices.4dbase*]. This folder stores all the elements needed for the proper functioning of the database.

Because of this, under Mac OS database folders now appear as packages. It is possible to double-click on the package to directly launch 4D, the database and the current data file. It is also possible to drag and drop the package onto the 4D application icon.

Under Windows, this change has no particular impact.

You can disable this default functioning by unchecking the **Create package for new databases** option in the Preferences, found on the "Application/Options" page:



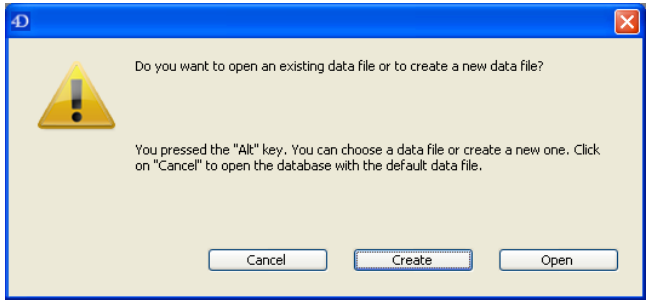
Changing the Data File

You can change the data file if you want (you can use the same structure with different data files).

In principle, you can use any data file with any structure file. It is not necessary for the data file to have the same name as the structure file, but the data it contains must be compatible with the structure that you want to use. Specifically, the data must correspond to the types of fields specified in the structure and the number of fields in the structure must be greater than or equal to the number of fields in the data file.

To change the data file, you can:

- When opening the database, choose the corresponding option in the “Data file” menu of the database opening dialog box. You can specify an existing file or create a new one (see the paragraph “Open Dialog Box Options” on page 40).
- During database startup, hold down the **Alt** (under Windows) or **Option** (under Mac OS) key. The following dialog box appears:



Click on **Open** to specify an existing file or on **Create** to create a new one.

- After the database is open, choose **New** > **Data File...** or **Open** > **Data File...** in the **File** menu of 4D or in the menu associated with the “New” or “Open” button of the 4D tool bar.

In all cases, a dialog box lets you select an existing data file or create a new one. If you choose to create a new data file, 4D will open the database with the original structure but with no records.

When you use a different data file or create a new one, it becomes the new current data file and will be used automatically on the next opening of the database.

If you more or rename the data file, it will be necessary to locate it manually.

Associating a Data File with a Structure

In most cases, you may not want for the user to be able to open a different data file. One important reason that may influence this decision is that if the structure file is not compatible with the data file, the data file will be reconfigured in order to correspond to the structure file. To avoid this, use the link signature (also called the WEDD signature) in order to only allow a data file to be opened with the suitable structure file. Locking a data file prevents it from being opened with a structure file that has a different signature. However, this does not prevent the structure file from opening a data file that does not have a signature. For more information about the link signature, please refer to the paragraph “WEDD” on page 199.

Installing Plug-ins

Plug-ins bring additional functionalities to your applications. For example, the 4D Write plug-in adds the functions of a word-processing software. Certain plug-ins such as 4D Pack are provided free of charge, others may require the purchase of a specific license.

Plug-ins fit into your 4D and 4D Server environment. In order to make use of them within your databases, you must install them in suitable locations. Various locations are possible depending on your mode of use of the plugins and your operating system.

Note *4D Components* are another means of adding additional functionalities to your applications. For more information, please refer to the chapter “4D Components” on page 1269.

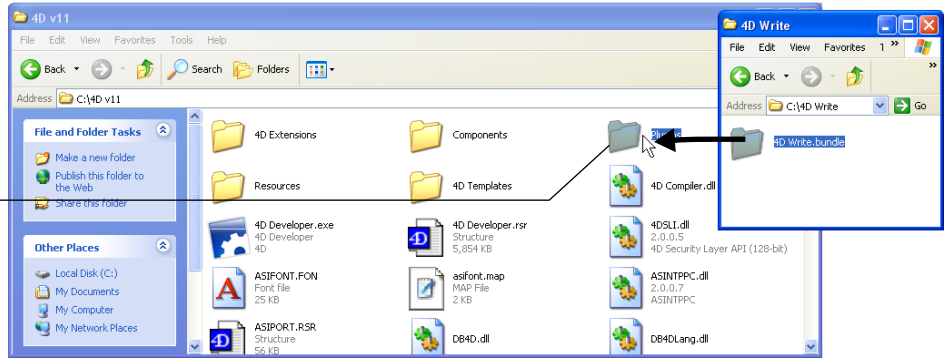
Installation Principles

Installing plug-ins in the 4D environment is done by copying the plug-in files into a folder named **PlugIns**. This folder can be placed in two different locations depending on your needs (see the section “Location of the PlugIns Folder” on page 49).

The “*PluginName*.bundle” folders (called packages under Mac OS) contain both Windows and Mac OS versions of the 4D plug-ins. More particularly, their specific internal architecture allows 4D Server to load the appropriate version according to the platform on which the 4D Client machine will be run.

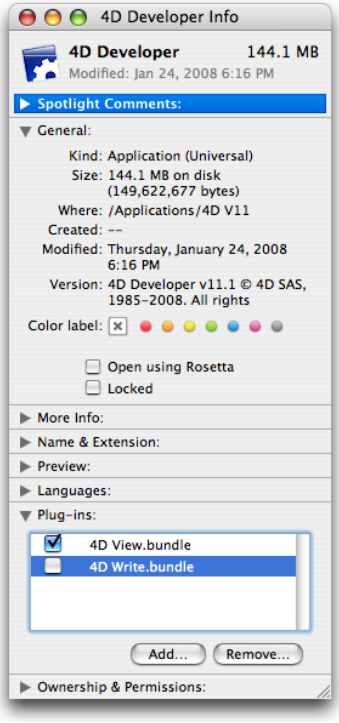
To install a plug-in in your environment, you just need to put the “*PluginName*.bundle” folder or package concerned into the desired **PlugIns** folder.

- Under Windows, this is done by simply copying the files:



Example of location: PlugIns folder placed next to executable application

- Under Mac OS, installing plug-ins can be done either by copying the files, or, even simpler (option recommended for plug-ins installed at the application level), using the “Plug-ins” section of the “Info” dialog box (see the section “Specific Management of Plug-ins under Mac OS” on page 50).



The plug-ins are loaded by 4D when the application is launched. You must therefore quit your 4D application before installing plug-ins.

Then open your database with 4D. If the plug-ins require a specific license for use, the plug-in will be loaded but not activated.

Location of the PlugIns Folder

You can place the **PlugIns** folder in two different places:

- At the level of the 4D executable application.
In this case, the plug-ins are available for all of the databases opened by this application.
Under Mac OS, if the PlugIns folder is placed in this location, it can be managed via the “Info” dialog box (see the following paragraph).

- At the same level as the database structure file.
In this case, the plug-ins are only available in this database.

The choice of location will depend on your mode of using the plug-in.

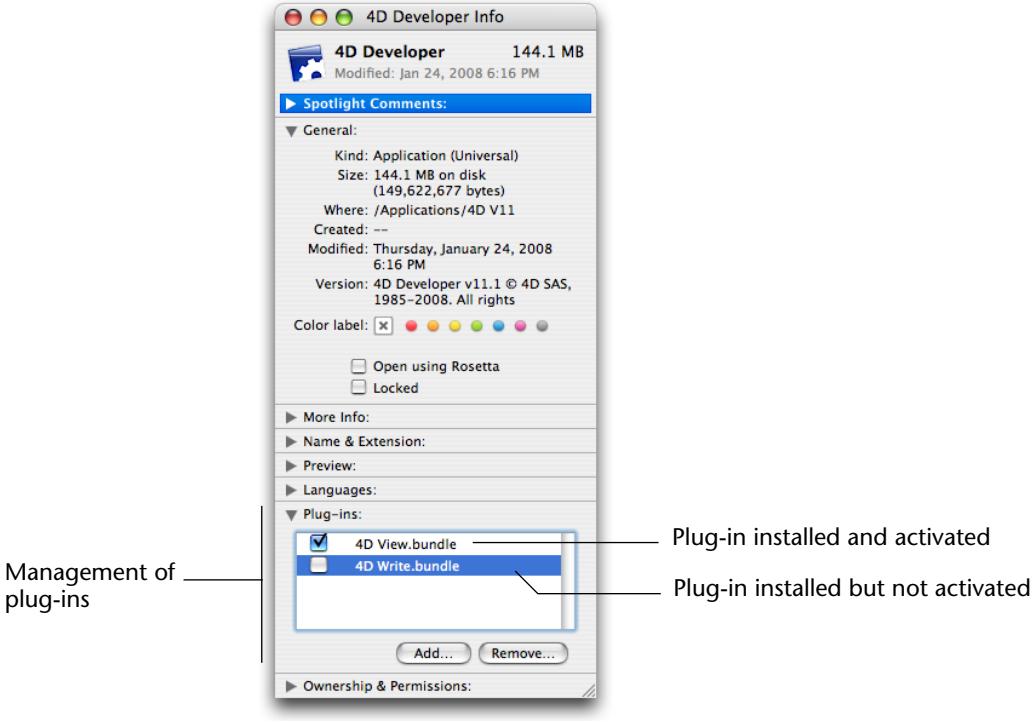
If the same plug-in is placed in both locations, 4D will only load the plug-in located next to the structure.
In the case of an application compiled and merged using *4D Desktop Volume*, the presence of several instances of the same plug-in will prevent the application from opening.

Specific Management of Plug-ins under Mac OS

Under Mac OS, plug-ins installed at the level of the 4D application (**PlugIns** folder placed in the package) can be managed via a specific system interface, without it being necessary to handle the package. This interface is found in the "Info" dialog box of the 4D application, which can be accessed at the level of the Finder.

To display the "Info" dialog box, **Ctrl+click** on the 4D or 4D Server icon and choose **Get Info** in the contextual menu.

The "Plug-ins" area is used to manage the plug-ins:



- To add a plug-in, click on **Add** and specify the plug-in package to install.
- To delete a plug-in, select its name in the list and click on **Remove**.
- Each plug-in can be deactivated or reactivated via the associated check-box; a deactivated plug-in is not loaded by the application but it is not removed from the folder.

Note The 4D application must not be launched while you are adding or removing plug-ins.

Plug-ins and Universal Binary architecture

Under Mac OS, plug-ins must be of the Universal Binary architecture so that you can execute 4D in this mode and take advantage of its performance. If this is not the case, you can still use the plug-ins in 4D in Rosetta mode (see the paragraph "Universal Binary Architecture (Mac OS)" on page 32).

Converting Databases from Previous Versions

Databases created with versions 6.x, 2003.x and 2004.x of 4th Dimension or 4D Server are compatible with version 11 or higher (structure and data files). However, these databases must be converted to version 11 and can no longer be opened with their original version afterwards.

Structural changes made at the level of the database engine of 4D v11 require in-depth conversion of both the structure and data. This conversion is carried out via a specific wizard (see the paragraph below). By precaution, the wizard makes a copy of the original database before converting it so that you can revert back at any time.

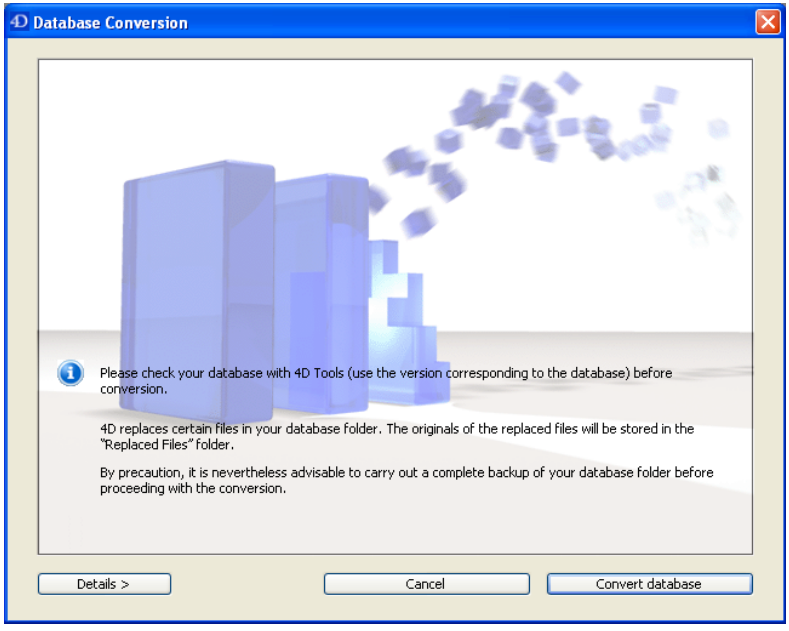
Moreover, certain obsolete or former mechanisms are no longer supported and will be removed or replaced during conversion (for more information about this point, please refer to the *4D v11 SQL Upgrade* manual.

General Case

You can convert any interpreted structure file. The file may contain compiled code; in this case, it will be necessary to recompile the database after conversion.

It is recommended to check the integrity of the database before conversion using the 4D Tools utility that corresponds to the version of the database (compacting, checking and, if necessary, repairing the database). Otherwise, if an anomaly is detected during conversion, the procedure will be stopped and the conversion wizard will ask you to use 4D Tools.

To convert a previous database, simply select it in the opening dialog box of 4D (see the paragraph “Opening a Database via the Open Dialog Box” on page 39) and click on **Open**. The conversion wizard will appear automatically:



Click on **Convert database** to start the standard procedure for converting the database and its data file. You can also display or modify the default parameters used for conversion by clicking on the. **Details >** button (see the following paragraph).

As a precaution, the conversion wizard will systematically make a complete copy of the database (structure and data) before beginning the procedure. If the conversion fails, you will thus still be able to recover your original database intact. Duplicating the database requires additional disk space. If the disk space available is insufficient, a warning message will appear and the conversion will be stopped.

The original files are copied into a folder named **Replaced Files (conversion)**, which is created next to the original files. If the data file is in the same folder as the structure file, the **Replaced Files (conversion)** folder will contain both original files.

If the data file is in another folder or on another volume, the conversion will create a **Replaced Files (conversion)** for each file.

Note The current log file of the converted database is also copied into the **Replaced Files (conversion)** folder and a new blank log file is created.

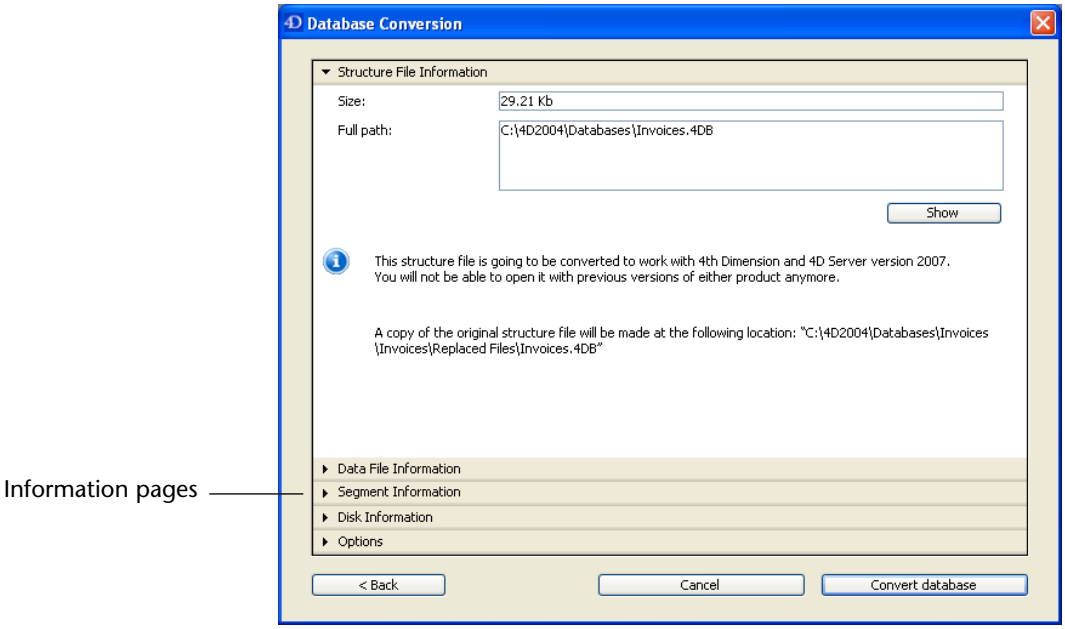
If the wizard encountered any non-critical errors during the conversion, they will be recorded in a conversion log file named *DataConversion_Log.log* that will be placed next to the converted structure file.

Note The user preferences set for the debugger (break points, saved expressions, etc.) are reinitialized during conversion.

View or Change Conversion Parameters

If you want to view or change certain parameters before carrying out the conversion, click on the **Details >** button. The dialog box will then display various information concerning the conversion parameters.

You can open the different pages of information by clicking on the corresponding areas:



- **Structure File Information**: This page shows the size and location of the original database structure file as well as the future location of the precautionary copy that will be made of this file.
- **Data File Information**: This page shows the size and location of the original database data file as well as the future location of the precautionary copy that will be made of this file.
  It can also be used to specify the data file to be converted. By default, the current data file is selected. You can either create a new blank data file (**Create a new data file** option) or choose to convert another data file by clicking on the **Choose another data file...** button.
  For more information about converting several data files, please refer to the section “Conversion of Databases Using Several Data Files” on page 55.
  The **Change the destination...** button lets you modify the location of the new data file after conversion.
- **Segment Information**: This page lists the segments of the data file associated with the database. For more information about this, please refer to the section “Conversion of Databases with Multiple Segments” on page 56.

- **Disk Information**: This page indicates the space available on the disk of the database.
- **Options**: This page contains a conversion option:
  - **Do not execute code when opening database**: When this option is checked, any code normally executed on database startup is deactivated when the converted database is launched for the first time. This option applies to code that is called using the On Startup database method. This allows you to avoid initialization errors that may occur following conversions.

Files of Converted Databases

After conversion, in addition to the *.4DIndy* and *.4DIndx* index files, 4D databases will contain the following additional elements:

- **DataConversion_Log.log**: This file stores any anomalies encountered during database conversion.
- **Replaced Files (conversion)** folder: This folder contains the copies of the original database files (structure, data and log) before conversion.

Specific Cases

Conversion of Databases Using Several Data Files

If the database that you want to convert uses several different data files, the conversion of the database and each of the data files will be carried out separately. The conversion of additional data files is carried out using the **Open>Data File** command of the **File** menu.

1 **Converting a database with its current data file.**

This procedure is described in the previous paragraph.

2 **In the converted structure, choose the Open>Data File... command in the File menu or in the menu associated with the "Open" button of the 4D tool bar.**

A standard Open document dialog box appears so that you can indicate the data file that you want to convert.

3 **Select the data file to be converted and click on Open.**

The conversion wizard window then appears. The information concerning the data file is updated depending on the file selected. The data file to be converted will first be duplicated and placed in the **Replaced Files (conversion)** folder, according to standard conversion principles.

4 **Validate the wizard window in order to launch the conversion.**

After the conversion is finished, the converted data file becomes the current data file of the database.

Conversion of Databases with "Previous Generation" Components

It is not possible to convert a structure file containing a previous generation component (prior to version 11). It is imperative to uninstall any components using the corresponding version of 4D Insider (utility available up to version 2004) before proceeding with the conversion of a structure file.

For more information about "new generation" components, please refer to the chapter "4D Components" on page 1269.

Conversion of Databases with Multiple Segments

In 4D v11 and higher, the size of the data file is now virtually unlimited (except for the limits imposed by the operating system). It is thus no longer possible to create or use data segments.

When converting a database that contains segments, the conversion wizard groups the contents of all the segments into a new data file. You must make sure that you have a sufficient amount of disk space for the new data file.

The segments of a database can be seen on the "Segment Information" page of the conversion dialog box (see the section "View or Change Conversion Parameters" on page 53).

If a segment is missing during conversion, the conversion wizard will let you indicate its location manually. If the segment cannot be located, conversion cannot take place.

Conversion of Databases with Multilingual Fields

When converting a database to version 11 and higher, textual data are converted into Unicode. In order for the conversion to be carried out correctly, 4D must know the source character set. By default, the character set corresponding to the current system language is used. In previous versions of 4D, it was possible to associate a specific character set to form objects via the **Keyboard Layout** property. In this case, 4D v11 is not able to deduce the source character set with certainty since the property was stored with the form and data entry could be carried out via intermediary variables.

It is up to the developer to explicitly declare the specific character sets used **before conversion**, via a text file placed at the same level as the data file to be converted. This file must have the following characteristics:

- *Name*: **multilang.txt**
- *Encoding*: **ANSI** or **Mac Roman** (do not use Unicode)
- *Format*: **table_number separator field_number separator subfield_number (optional) separator dialect_code CRLF**
  - The separator is ";".
  - Each row must be ended with a carriage return (CR or CRLF), empty rows and spaces are allowed.
- *Example*:
  Given a database with 4 tables (TABLE_1 to TABLE_4).
  - In TABLE_3, there are 4 Alpha fields (field_1 to field_4).
  - In TABLE_4, there are 6 Longint fields and a SUBFIELD subtable with 2 Alpha fields
  - The "Keyboard Layout" property (dialect) has been modified so as to be able to enter Greek in [TABLE_3]field_3 and [TABLE_4][SUB-FIELD]field_1, and Russian in [TABLE_3]field_4 and [TABLE_4][SUB-FIELD]field_2
  - The *multilang.txt* file must therefore contain:
  3;3;1049
  3;4;1032

  4;7;1;1049
  4;7;2;1032

  The dialect codes are described in the *keyboardmapping.xml* file, located in the *4D Extensions* subfolder.

2 Overview of Design Environment

The purpose of this chapter is to describe the 4D Design environment, its interface and how it works. The following subjects are covered:

- A description of the different 4D working environments,
- An overview of the Design environment editors you use to create a database,
- An overview of the integrated data management tools,
- An overview of the compiler and the application builder,
- An overview of the Web Services Wizard,
- Instructions for navigation among the menus and windows of 4D,
- Instructions for using the 4D Explorer,
- Instructions for using the 4D Runtime Explorer,
- Instructions for using the Search editor of the Design environment,
- Instructions for configuration and use of drag and drop functions,
- A description of the specific features related to 4D Server.

Unless otherwise noted, all instructions and explanations apply to both 4D and 4D Server.

Working Environments

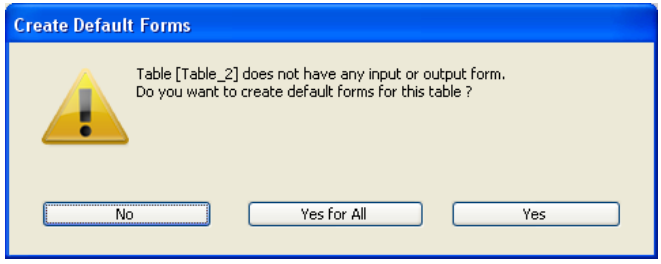
4D can be used in two distinct environments: the Design environment and the Application environment.

The Design Environment

The Design environment is used to design, develop and test your databases. All aspects of your database design are implemented in the Design environment. You use the Design environment to create tables and fields; define relations among tables; create forms for data entry, display, and printing; implement a password access system; create custom menus; or attach methods to database objects.

For example, you might want to keep track of information about each of the employees in a company. In the Design environment, you create an [Employees] table and add fields to that table to store employee data, such as the employee's name, job title, start date, and salary. You might also add a [Departments] table that contains information about each department in the company. You could then create a *relation* between these tables that lets you easily determine in which department an employee works and which employees work in each department.

The Design environment provides access to various windows and editors that allow you to carry out standard operations on the data of your database. You can enter data, search for a particular record, import or export data, print reports or generate labels for a mail shot. This means that you can test the functioning of your database, for example by entering or importing some records, or by executing your methods. The editors provided are the same as those available to users in the Application environment.
By default, the first time you attempt to view the data, 4D offers to automatically create the forms required for your database, if necessary:



You can use the Design environment to do the following:

- Create tables and fields in which to store data,
- Establish relations between tables,
- Create forms for entering, displaying, printing or publishing data,
- Create lists of choices that simplify and control data entry,
- Write and execute methods to automate database operations,
- Create custom menus and associate methods or automatic standard actions to them,
- Create and manage multiple processes, allowing you to perform multiple database operations at the same time,
- Set up a system of passwords to control access to information,
- Publish a database as a Web server or as an SQL server,
- Call or publish Web Services,
- Compile the database in order to accelerate its execution,
- Merge the database and the 4D engine in order to build a stand-alone application.
- Enter and modify data,
- View and print data,
- Search and sort records,
- Create reports, labels and graphs,
- Import and export data,
- Work with any 4D plug-ins installed in the database.

Each of these features is described in this manual.

The Application Environment

The Application environment is the environment that you use to run a custom application — an application that uses 4D but has its own menu system and screen design. This is the environment in which applications created in the Design environment are usually distributed. To access it from the Design environment, you just need to select **Test Application** in the **Run** menu.

Note • The Application environment can be used only if at least one menu was created using the Menu Bar editor. By default, 4D creates a standard menu bar.
• It is possible to automatically close all the windows of the Design environment when switching to the Application environment (see the "Options Page" paragraph on page 163).

In the Application environment, you control everything in the application, from the menus and forms it uses, to the methods used to accept, process, and display data. You are responsible for providing menu items and associated methods or standard actions that manage basic tasks such as data entry and modification, searching and sorting, and reporting. You can use any or all of the standard editors provided in 4D (order by, label, etc.) or create your own screens and editors.

The Application environment can be completely different for each application you create. From the user's standpoint, the Application environment is a complete application for a specific kind of information management.

The default menu bar generated by 4D includes a **Mode** menu allowing you to "Return to Design mode."

Note If you have windows from more than one environment open at the same time, you can switch between environments by clicking their respective windows. When you click on a window, 4D places this window in the foreground and makes it the active environment.

Design Environment Editors

4D editors are used to create and modify the various components of your database design. Each editor is dedicated to one aspect of the design.

The Design environment contains the following editors:

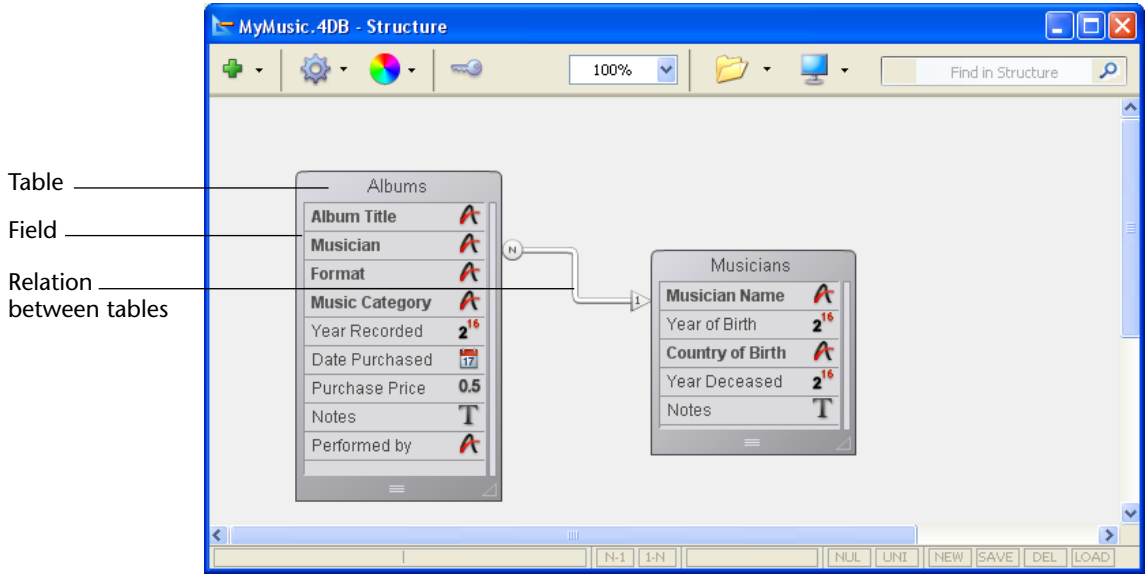
- Structure editor
- Form editor
- Method editor

- Tool Box.
  The tool box groups several secondary editors, used to manage specific aspects of the database, in the same environment:
  - Users and user groups,
  - Menu Bar editor,
  - Picture library,
  - Help Tip editor,
  - Lists editor,
  - Style Sheet editor,
  - Filter and Format editor.

  The integrated data management editors are described in the "The Integrated Data Management Tools" paragraph on page 78.

The Structure Editor

The Structure editor is your starting point for all design operations. Use the Structure editor to create tables, fields, and relations between tables. The Structure editor displays the images of the tables in the database and graphically shows the relations (if any) among the tables. The following illustration shows the Structure editor window.



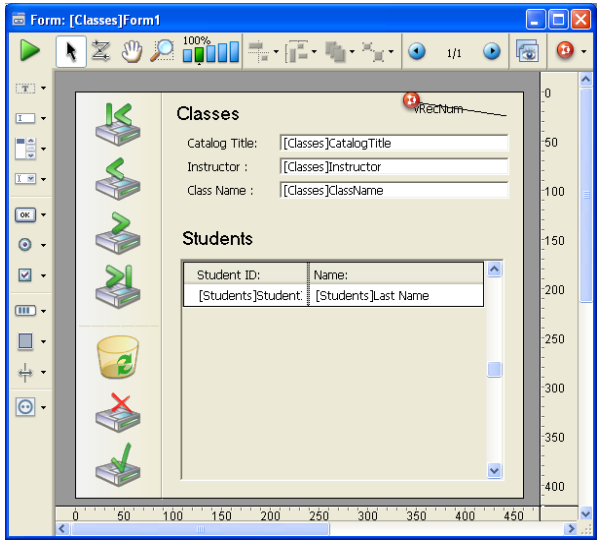
Use the Structure editor to:

- Create and edit tables and fields,
- Specify table and field properties,
- Relate tables and set relation properties,
- Create and manage indexes,
- View and analyze the structure of a database.

See the chapter “Database Structure” on page 239 for more information about using the Structure editor.

The Form Editor

Use the Form Wizard to create forms and the Form editor to modify forms — both on-screen forms and printed reports. The Form editor lets you create sophisticated forms for managing your data.



Use the Form editor to:

- Add fields and variables to a form,
- Add buttons, combo boxes, drop-down lists, tab controls, and other interface objects to forms,
- Specify display formats and entry filters for data displayed and entered on the form,
- Add methods that enforce business rules during data entry or manage interface elements,

- Add graphic objects to the form — including text, lines, rectangles, and ovals — or paste in pictures or graphics that were digitized or created in another application,
- Specify fonts and font styles for objects containing text,
- Specify drag-and-drop actions,
- Specify automatic resizing and repositioning actions,
- Establish access privileges for forms.

Refer to the chapters "Creating Forms" on page 335 and "Form Editor" on page 371 for detailed information on how to create a form with the Form Wizard and how to modify forms with the Form editor and the object libraries. Refer to the chapter "Fields and Active Objects in Forms" on page 483 for information about using the Form editor to control data entry and add interface elements to data entry forms. Refer to the chapter "Output Displays and Reports" on page 635 for information about using the Form editor to design output forms, reports, mail-merge documents, and forms for mailing labels.

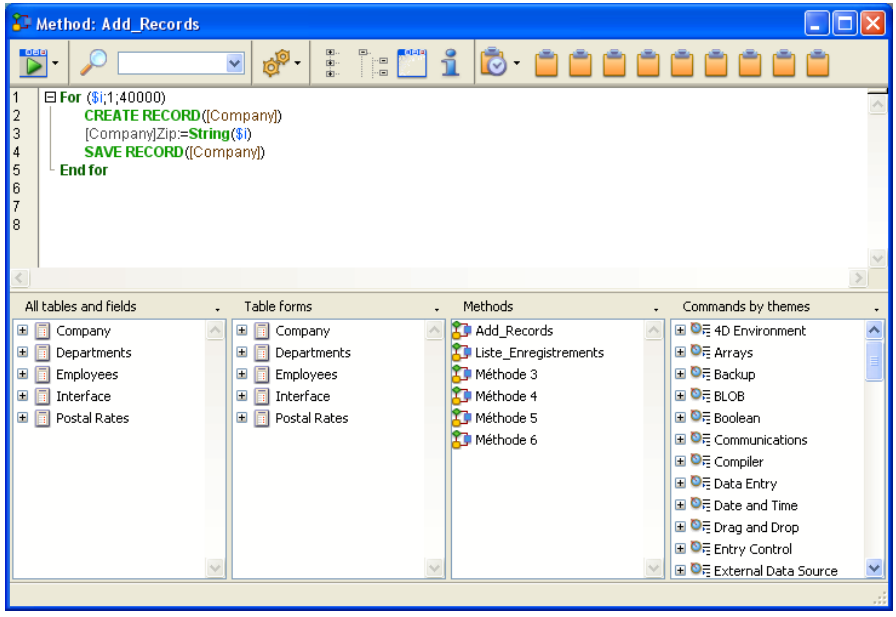
The Method Editor

Methods are instructions that process data or perform actions. Methods can perform tasks such as calculating the values of fields, transferring data from one table to another or validating data as it is entered. Methods also perform actions such as sorting, displaying or printing records.

4D allows the use of two languages in the Method editor:

- The 4D language, similar to Pascal, which includes a set of commands allowing all types of programming for the interface and functions.
- The SQL language, a standard language geared towards database management.

Here is a window of the Method editor:



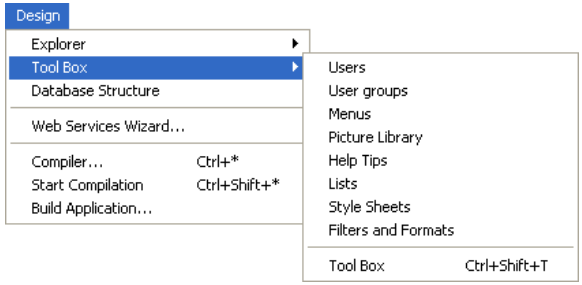
You can use the Method editor to:

- Write database methods that run automatically when particular worksession-related events occur,
- Write triggers that run automatically when particular database engine events occur,
- Create form methods that run automatically when a form is used,
- Create project methods that can be attached to custom menus, called by other methods, or executed manually,
- Create object methods that are associated with fields or other form objects.

Refer to the chapter “Methods” on page 757 for more information about how to use the Method editor. Refer to the *4D Language Reference* manual for detailed information about writing methods and to the *4D SQL Reference* manual for more information about writing SQL code.

The Tool Box

The tool box groups together various secondary editors of 4D. It can be accessed using the hierarchical **Tool Box** command in the **Design** menu:

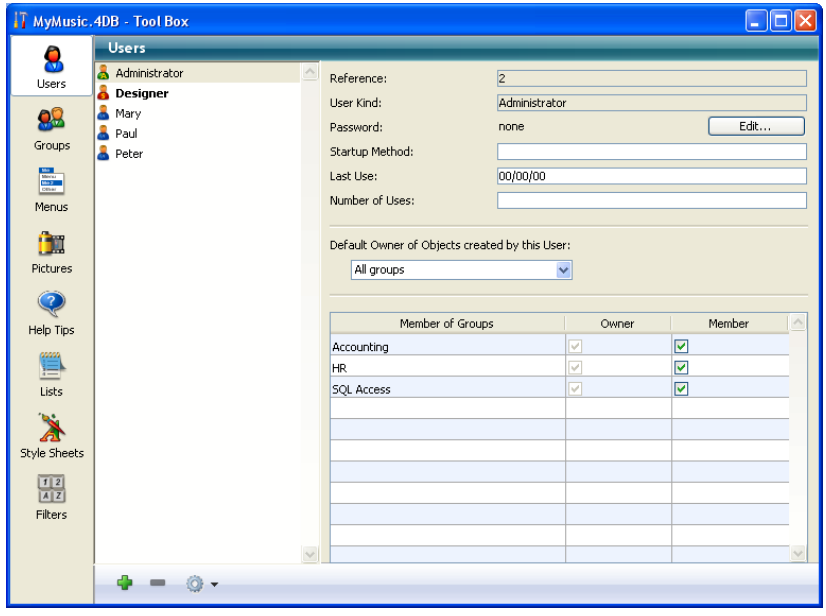


Each sub-command displays the corresponding page of the tool box. The **Tool Box** command opens the tool box to the current page. You can also use the **Tool Box** button of the toolbar.

By default, the tool box displays the first page (Users).

The Users and Groups Editors

4D lets you create a system of users and groups in order to control access to different parts of your database and application. The screen below shows the User editor in the 4D tool box:



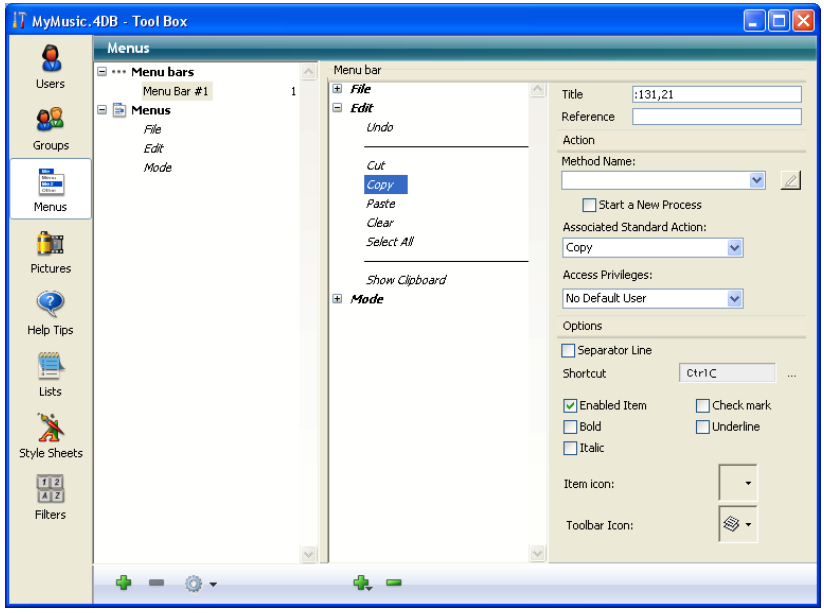
The Users and Groups editors can be used to:

- Create users and give them passwords,
- Allow specific users to add users and change user passwords,
- Place users in groups,
- Give groups access to parts of the database, such as particular forms, menu commands, methods, and plug-ins,
- Monitor database use by individual users.

Refer to the chapter “Users and Groups” on page 831 for more information about the access control system of 4D using users and groups.

The Menu Bar Editor

When you create custom applications with 4D, you use the Menu Bar editor to create menu bars, menus, and menu commands. You can also attach menus to any form that you use for data entry. The figure below shows the Menu Bar editor being used to create a menu bar in the tool box of 4D.



Use the Menu Bar Editor to:

- Create menu bars,
- Create and modify custom menus and menu commands,

- Attach icons to menu commands to create a custom toolbar,
- Assign project methods or standard actions to menu commands,
- Attach submenus to menu commands,
- Include graphics for splash screens that display with menu bars,
- Establish access rights to menu commands,
- Set keyboard shortcuts for menu commands,
- Enable or disable menu commands,
- Start a new process from a menu command.

Refer to the chapter “Custom Menus” on page 859 for a detailed discussion of adding custom menus and menu bars to your applications.

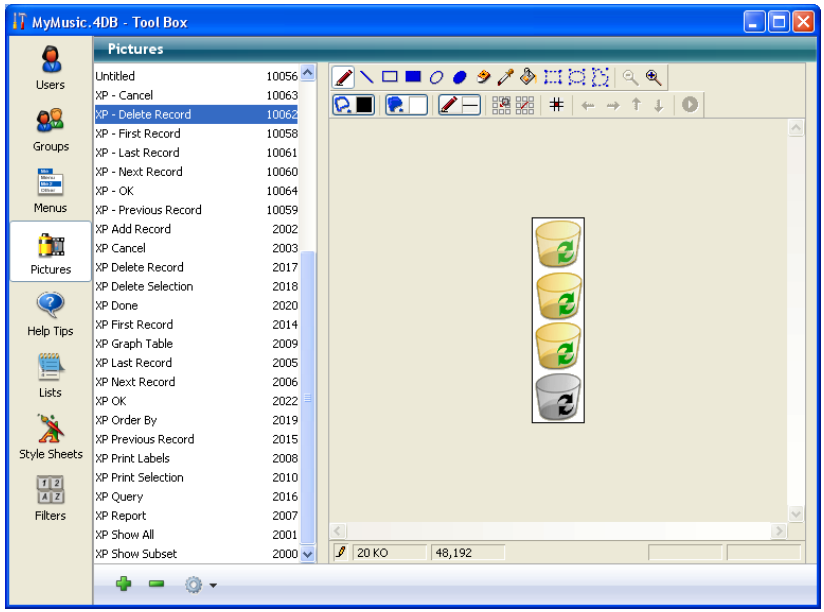
The Picture Library

Use the Picture Library to store graphics that you can use as design elements in forms, as toolbar or list icons, picture menu items, or picture buttons. With the Picture Library, you can use a graphic in several places in your database but you need to store it in only one place. When you update a picture in the Picture Library, all references to the picture are updated automatically. This feature can reduce the size of your Structure files and make changes to the database easier to manage.

The Picture Library also includes a 2D Paint editor that allows you to create or retouch pictures. It is an ideal environment to design buttons and icons.

In addition to the features mentioned above, the Picture Library also includes integrated functions that can create or edit row by column arrays of pictures that are used to create picture buttons or picture menus.

The Picture Library is shown below as it appears in the tool box of 4D:



The Picture Library is used to:

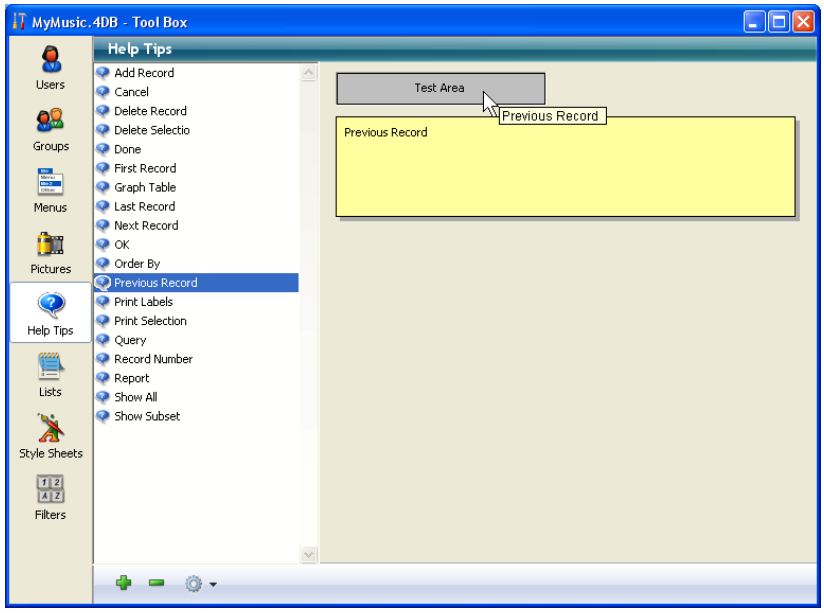
- Store and preview the pictures of the database structure,
- Create or import new pictures,
- Retouch pictures,
- Define and preview tables of thumbnails,
- Drag and drop pictures to the Form editor, the Lists editor, or the Menu Bar editor,
- Delete pictures.

For more information about the Picture Library, refer to the chapter "Picture Library" on page 891. For more information on inserting a picture in the Form editor, refer to "Inserting a Static Picture in a Form," page 453, "Picture Buttons," page 566, "Picture Pop-up Menus," page 579, "Toolbar Icons," page 885 and "Adding a Small Icon to an Item," page 913.

The Help Tip Editor

The Help Tip editor lets you create help messages that can be displayed as tips when the mouse moves over form objects. Associating a help tip with an object is carried out in the Form editor.

The screen below shows the Help Tip editor in the tool box of 4D:



The Help Tip editor is used to:

- Create and display the help tips intended for users.
- Test whether the help tips appear as desired.

For more information about using the Help Tip editor, refer to the "Adding Help Tips to a Field or Object" paragraph on page 517.

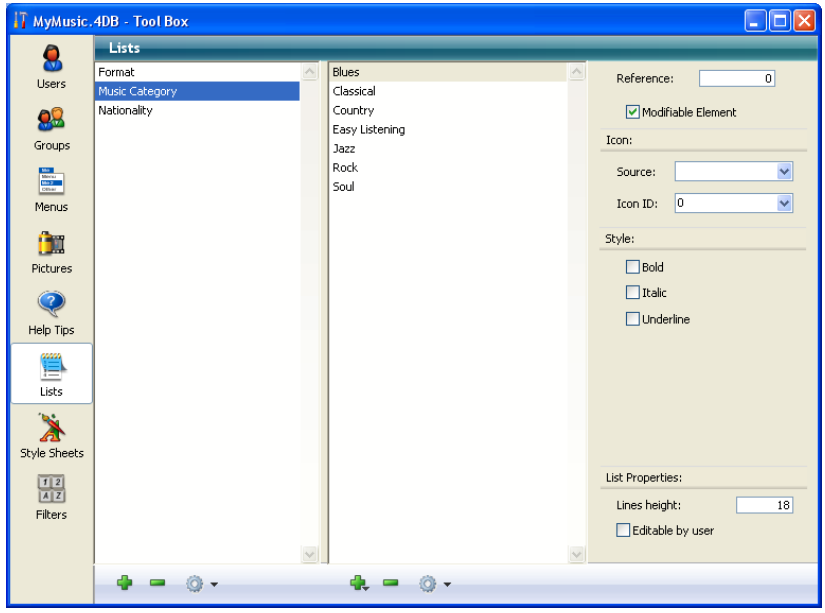
The Lists Editor

You use the Lists editor to create lists. Lists can be used for several purposes in a database. Here are the most common uses:

- You can attach a list to a field. A user can select an entry from a choice list instead of typing it. With a choice list, you prevent the entry of misspelled words and incorrect data.
- You can use a list to specify the items in pop-up menus, drop-down list boxes, combo boxes, scrollable areas, or tab controls.
- You can create hierarchical lists to populate hierarchical list objects, hierarchical pop-up menus, or tab controls. These objects are then automatically defined using the list values.

- You can access the items of a list in your methods or transfer the elements of an array to a list (and vice versa). By doing so, you can use a list to populate pop-up menus, drop-down lists, list boxes, combo boxes, or scrollable areas.

  The figure below shows the Lists editor in the tool box of 4D being used to create a list:



  Use the Lists editor to do the following:

- Create choice lists,
- Add items to a choice list,
- Attach small icons to list items,
- Delete choice lists or individual items in a list,
- Sort choices in a list,
- Activate values for use in tab controls,
- Make values editable for use in a hierarchical list,
- Make a choice list modifiable by the user.

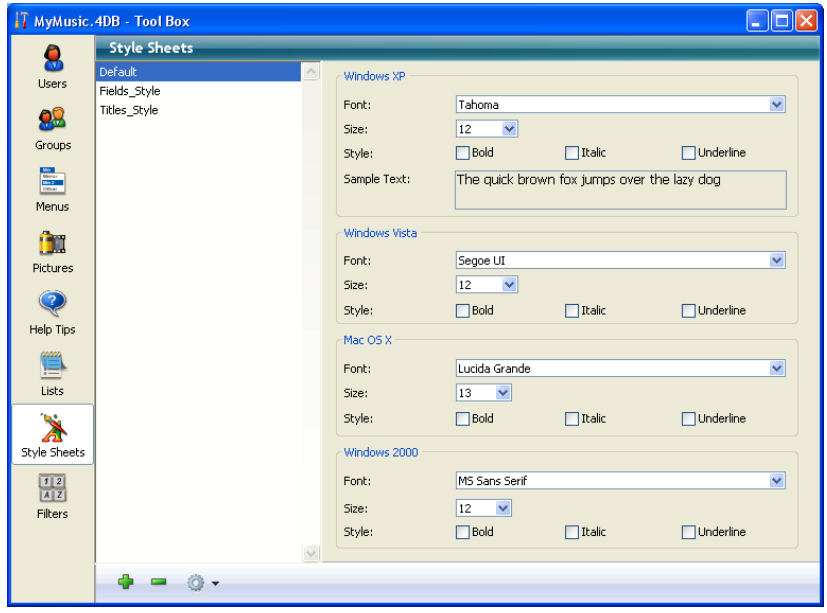
  Refer to the chapter “Lists” on page 903 for a detailed discussion of lists.

The Style Sheet Editor

The Style Sheet editor lets you save different combinations of font attributes — the character font, its size and its style — in the form of style sheets. The style sheets defined can then be used to set the font attributes in the Form Wizard or Form editor.

4D style sheets can contain a specific definition for each platform (Windows XP, Windows Vista, Windows 2000 and Mac OS X), which makes it easier to set up multi-platform interfaces.

The following screen shows the Style Sheet editor in the tool box of 4D:



Refer to the chapter “Style Sheets” on page 923 for more information about the use of style sheets.

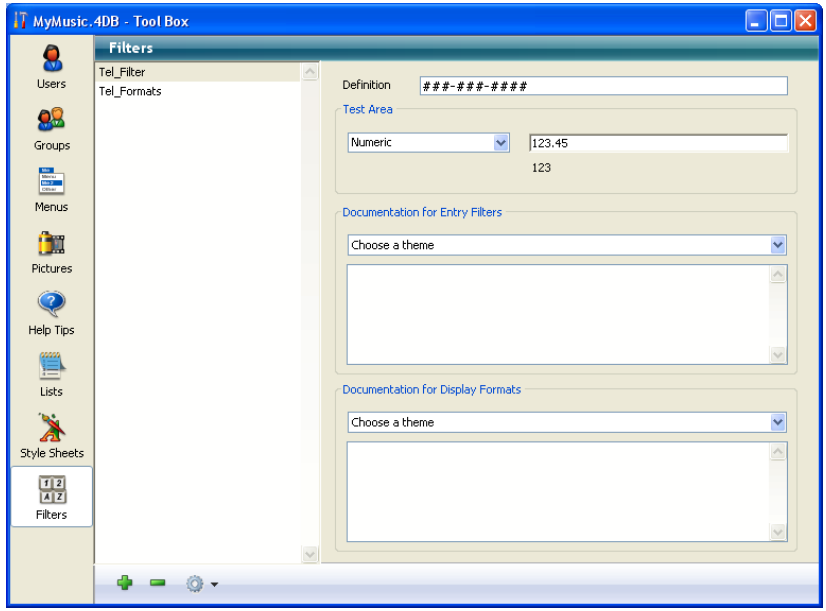
The Filter and Format Editor

The Filter and Format editor lets you create entry filters and display formats that can be associated with any enterable object or used for display purposes in your forms.

An entry filter controls exactly what the user can type when entering data. For example, if a component number always consists of three letters followed by three numbers, you can restrict the entry to respect this form. You can apply an entry filter to a field or an enterable object.

A display format lets you put the data of a field or variable into shape before displaying it.

The following screen shows the Filter and Format editor in the tool box of 4D:



For more information about using filters and formats, refer to the "Creating Custom Display Formats and Entry Filters" paragraph on page 503.

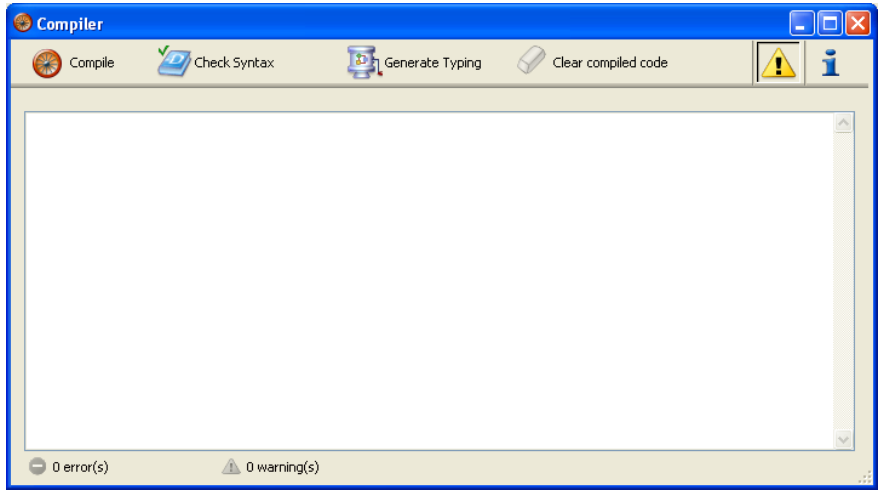
Compiler and Application Builder

Compiler

4D has an integrated compiler that can translate the entirety of the database code into machine language. Compiled databases run more quickly because 4D does not have to interpret the code each time a method is executed.

Note For more information about the principles of compilation, refer to the "What is a Compiler?" paragraph on page 1229.

Compilation is configured and executed from the “Compilation” dialog box that can be accessed using the **Compiler** command of the **Design** menu or the corresponding button in the 4D tool bar.



Note It is also possible to execute compilation directly using the **Start Compilation** command of the **Design** menu.

This window is used for:

- Launching the compilation of the database (only with 4D Developer),
- Checking the syntax of the methods,
- Building/rebuilding database typing methods,
- Deleting the compiled code,
- Displaying or hiding the *warnings*,
- Accessing the compilation preferences.

Notes
• Database compilation requires an appropriate license. Without this license, it is not possible to carry out a compilation (the menu command is titled **Check Syntax** and the **Compile** button is disabled in the window). Nevertheless, it is still possible to check the syntax and generate typing methods.
• With 4D Client, only the **Check Syntax** and **Generate Typing** buttons are active.

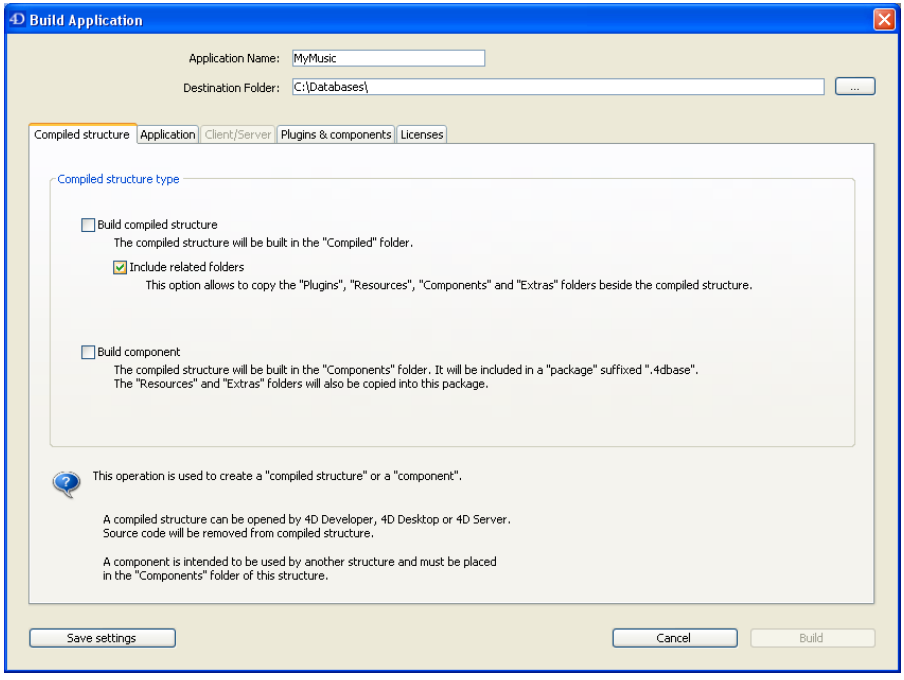
Once the database has been compiled, you can choose the running mode — compiled or interpreted — on startup of the database (see the "Opening an Existing Database" paragraph on page 38) or via the **Run** menu (see "The Run Menu" paragraph on page 90).

For more information about the compiler, refer to the chapter "Compilation" on page 1229.

Application Builder

Once a database has been compiled, you can use the **application builder** to create either a compiled version of the database (without interpreted code), or a double-clickable version integrating 4D Desktop Volume, the 4D database engine.

To open the window used for building applications, choose the **Build Application...** command in the **Design** menu. The following window appears:



Note If the database has not yet been compiled, an alert appears indicating that you must compile it.

The application builder allows you to:

- Build a compiled database (without interpreted code) from your database. A database without interpreted code is protected since it is impossible to read or modify the code.
- Build a component. A component is a structure containing functionalities that can be used by another structure (see the chapter “4D Components” on page 1269).
- Build a licensed executable application. An executable application integrates the structure of your database and 4D Unlimited Desktop, the 4D database engine,
- Build a custom client-server application,
- Manage the insertion of plug-ins, components and license numbers into applications that are built.

For more information about the application builder, refer to the chapter “Finalizing and Deployment of 4D Applications” on page 1251.

Web Services Wizard

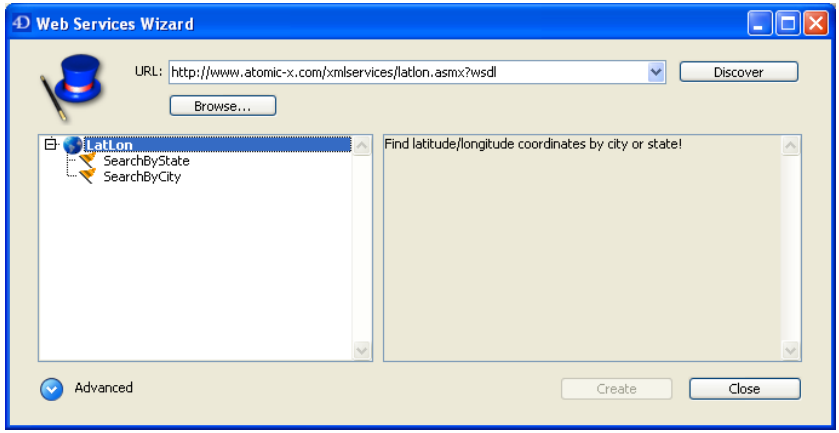
4D lets you publish or use Web Services within your databases.

A Web Service is a set of functions grouped together as an entity and published on a network such as the Internet. These functions can be called and used by any application compatible with Web Services and connected to a network. Web Services can carry out all sorts of tasks, such as supervising the routing of packages at a transporter’s, e-commerce transactions, monitoring market values, etc.

The program publishing the services is called the “server.” Any application compatible with Web Services can therefore use one or more of these functions; this is the “client” program. 4D can be used both as a Web Services client and/or server.

Note For more information about Web Services, refer to the chapter “Publishing or Using Web Services” on page 1137.

The Web Services Wizard is used for the “client” part. It allows 4D applications to connect to servers offering Web Services and to generate the project methods needed for interrogating these servers:



The Web Services Wizard allows you to:

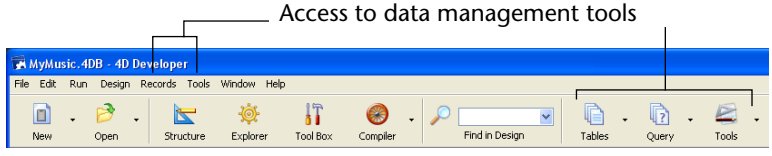
- Connect to a server providing Web Services,
- Parse and display the contents of the description file (WSDL),
- Generate the “proxy method” needed for connecting to and interrogating the server of the Web Services.

For more information about the Web Services Wizard, refer to the “Subscribing to a Web Service in 4D” paragraph on page 1149.

The Integrated Data Management Tools

The Design environment provides access to several editors and tools that can be used to create, display, edit, delete, sort, search or even print the data of your database. These editors can be used to test the functioning of the application or to carry out standard data management tasks.

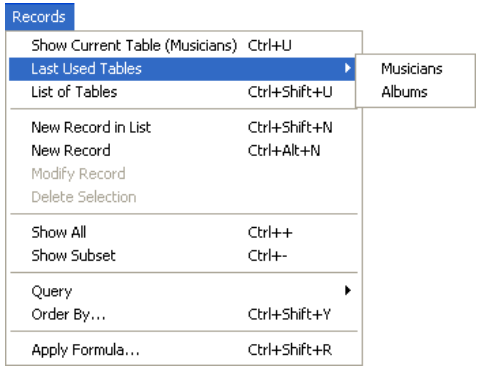
These editors can be accessed from the **Records** and **Tools** menu and using the **Tables**, **Query** and **Tools** buttons of the tool bar:



Record Display Window

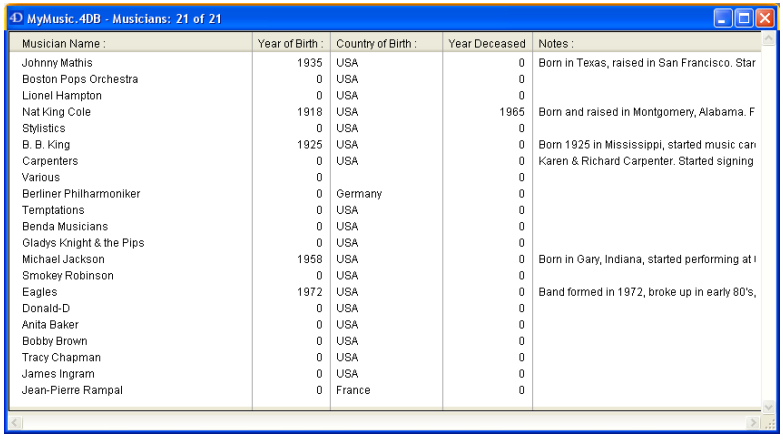
The record display window shows the list of records for a table in the current output form and can be used to access all the data management functions of the Design environment.

To open it, select the **Show Current Table (*Table Name*)** command in the **Records** menu or choose a table name from the **Last Used Tables** sublist.



Note These commands are also available via the **Tables** button of the tool bar.

This window can display a default form generated by 4D or by the Form Wizard, or a custom form built using the Form editor:



From this window, you can:

- View and edit values of the records in the database,

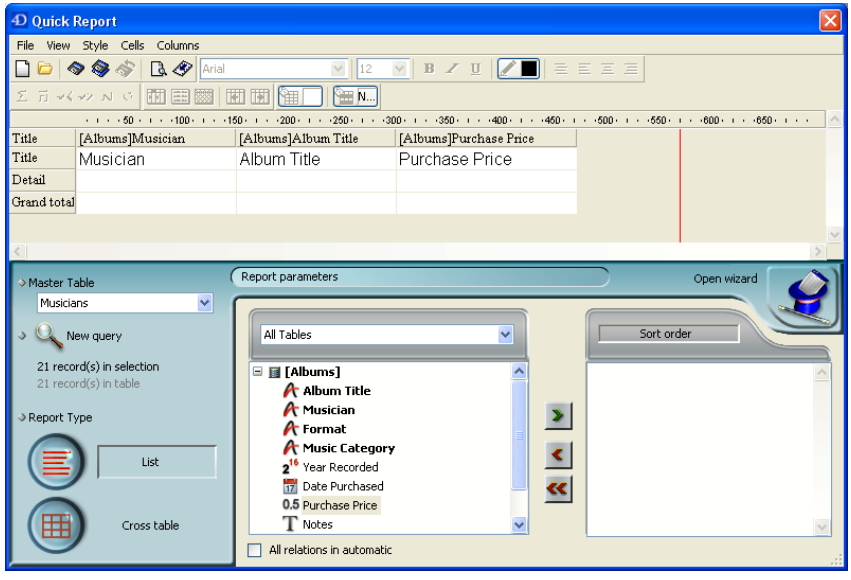
- Display the values of a record in a current page form by double-clicking a row,
- Create or delete records,
- Sort the records using the Order By editor,
- Select records manually or using the Query editor of 4D,
- Make overall changes to the records using the Formula editor.

For more information about the record display window and the integrated Order By, Query and Formula editors of 4D, please refer to the chapter “Record Management” on page 947.

Quick Report Editor

The Quick Report editor is one of two tools available in 4D for generating reports using your data — the other is the 4D Form editor, which can be used to build custom reports.

The Quick Report editor opens as in a separate window with its own menus and buttons:



It is possible to integrate a Quick Report window as an external area in a form.

Using the Quick Report editor, you can more particularly:

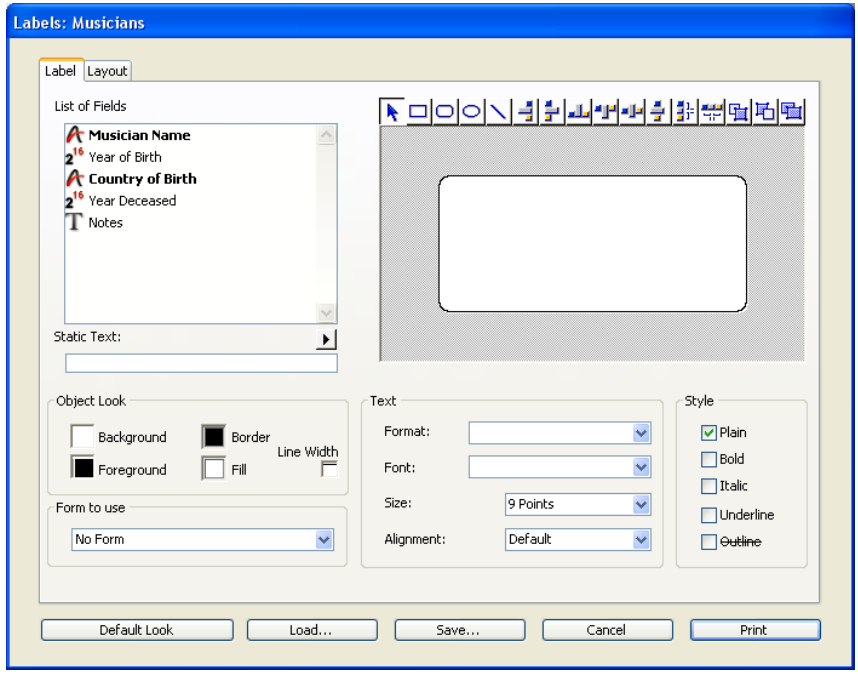
- Produce an ordered list of records

- Produce cross-table reports
- Compute summary calculations
- Modify fonts and styles in the report
- Define borders and background colors on a cell basis
- Save and open Quick Report designs to disk
- Select different output types such as HTML or text files, 4D View or 4D Chart areas, as well as printing or saving to disk.

For more information about the Quick Report editor, please refer to the chapter “Quick Reports” on page 699.

Label Editor

The 4D Label editor is a fast and simple means for creating and printing a wide variety of labels.



The Label editor lets you do the following:

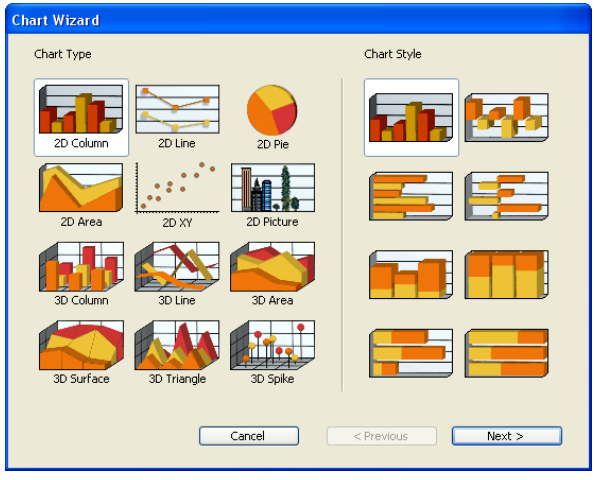
- Design labels for mail shots, catalogues, etc.,
- Create or insert decorative elements into a label design,
- Specify the font, font size and style to be used,

- Specify the number of labels across and down each page,
- Specify the number of labels to print for each record,
- Specify the label page margins,
- Indicate a method to be executed when printing each label or record,
- Load and save label designs,
- Print labels.

For more information about the Label editor, please refer to the chapter “Labels” on page 1031.

Chart Editor

The **Charts** command of the **Tools** menu displays the 4D Chart Wizard window. This wizard provides you will all the tools needed for the graphic representation of your data as columns, areas, sectors, etc. :



As with quick reports, it is possible to use the chart editor as a separate window or as an included area in a form.

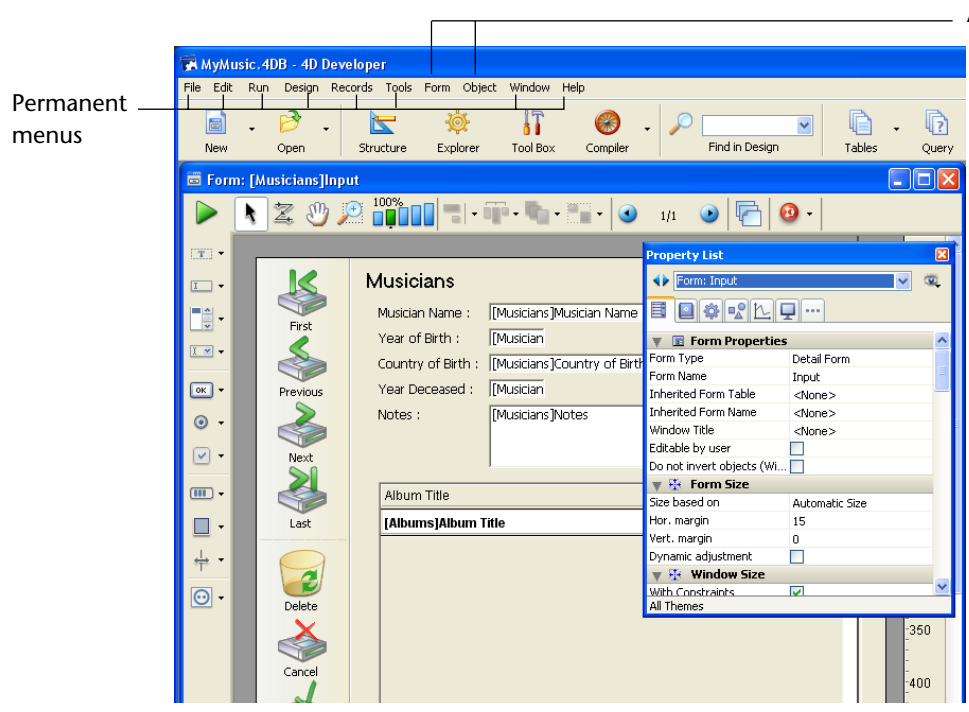
For more information about the Chart editor, please refer to the chapter “Graphs” on page 1053.

The Design Environment Interface

In the Design environment, you interact with 4D using its menus, contextual menus, toolbars, and windows. This section describes how to use these interface elements.

The Menus

In the Design environment, the first six menus starting from the left are always the same, no matter which editor you are using. Additional menus are added to the right of these permanent menus in the Form and Method editors.
The two rightmost menus, Window and Help, are also found in all the editors.
The screen below shows the menus added when the Form editor is opened.



When several editor windows are open, the menus belonging to the frontmost editor window appear. You choose menu commands from these menus as you would in any application. For complete instructions about choosing from a menu, see the documentation that came with your computer.
The following menus are permanent:

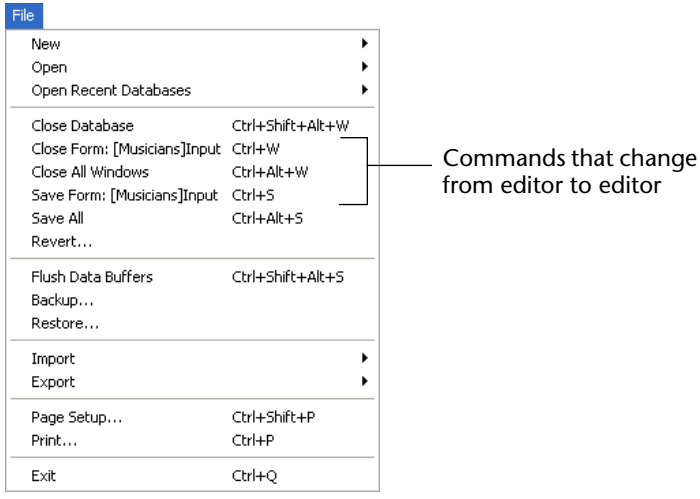
- File
- Edit
- Run
- Design

- Records
- Tools
- Window
- Help.

Note Under Mac OS, the application menu (named **4D Developer**, **4D Server** or **4D Client** according to the current application) is added to the left of the menu bar. It contains more particularly the **Preferences** and **Quit** commands, described below in the **Edit** and **File** menus.

The File Menu

The **File** menu of 4D provides various commands related to file management. A part of the menu varies according to the window open in the foreground.



The following commands are independent of the active editor:

- **New > Database...**: This command opens the standard save file dialog box, which can be used to create a new database.
- **New > Database From Template...**: This command opens the template selection dialog box (see the "Creating a Database from a Template" paragraph on page 34).
- **New > Database From Structure Definition...**: This command creates a database from an XML structure definition file (see the "Creating a Database from a Structure Definition" paragraph on page 334).

- **New > Data File...**: This command opens the standard save file dialog box, which can be used to create a new data file for the current database (see the "Changing the Data File" paragraph on page 46).
- **New > Object Library...**: This command displays the standard save file dialog box, which lets you create a new object library (extension .4IL). For more information about object libraries, refer to the "Using Object Libraries" paragraph on page 461.
- **New > Table...**: This command displays the table creation dialog box (see the "Creating and Modifying a Table" paragraph on page 256). When you validate this dialog box, the Structure editor moves to the foreground.
- **New > Form...**: This command opens the Form Wizard (see "The Form Wizard" paragraph on page 347).
- **New > Method...**: This command displays the project method creation dialog box (see the "Creating or Opening a Project Method" paragraph on page 766). When you validate this dialog box, the Method editor moves to the foreground.
- **Open > Database...**: This command opens a standard open file dialog box, which lets you open an existing database (see the "Opening a Database via the Open Dialog Box" paragraph on page 39).
- **Open > Data File...**: This command opens a standard open file dialog box, which lets you open another data file that can be used with the existing database (see the "Changing the Data File" paragraph on page 46).
- **Open> Object Library...**: This command displays the standard open file dialog box, which lets you open an object library (extension .4IL). For more information about object libraries, refer to the "Object Libraries" paragraph on page 347.
- **Open > Form...**: This command opens the Forms page of the Explorer (see the "Forms Page" paragraph on page 112).
- **Open > Method...**: This command displays the Methods page of the Explorer with the list of project methods expanded (see the "Methods Page" paragraph on page 117).

- **Open Recent Databases** > ***list of recently-opened databases***: This command displays a submenu listing all the databases opened on the machine. The **Clear Menu** command can be used to reinitialize the contents of this menu.
- **Close Database**: This command closes the current database. The 4D application remains launched but there is no longer any database open. You can then select a New or Open command from the **File** menu or the **Maintenance Security Center** command in the **Help** menu.
- **Close *Object: Object Name***: This command closes the window located in the foreground at any time. 4D saves the contents of the window before closing it. The title of this command varies depending on the editor open.
- **Close All Windows**: This command closes all the open windows.

Note Under Mac OS, the **Close All** command can be access from the **File** menu when you press the **Option** key. It then replaces the **Close *Object*** command.

- **Save *ObjectName: Object Name***: You can save the contents of the foreground window at any time without having to exit 4D. 4D automatically saves the contents of an editor when you close its window, change environments or exit the application.
  This command is dimmed when no modification has been made to the object since the last time it was saved.
- **Save All**: This command saves the contents of all the windows that are open.

Note Under Mac OS, the **Save All** command can be accessed from the **File** menu when you press the **Option** key. It then replaces the **Save *Object*** command.

- **Revert**: You can return to the last saved version of a form or method. This menu command replaces the contents of the Form editor or Method editor by that of the last saved version.

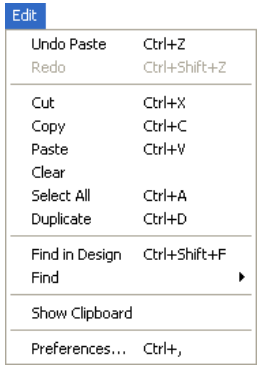
- **Flush Data Buffers**: This command “forces” the data cache to be flushed.
  Before being written to disk, the operations carried out on the data are kept in the data cache during a certain period of time. This permits faster data access. The cache is flushed at regular intervals (every 15 minutes by default) and whenever the database is exited. In certain cases, you may want for the cache to be flushed immediately. You can then use this command.
- **Backup...**: This command displays a dialog box that can be used to launch a manual backup of the database. For more information, please refer to the “Performing a Backup” paragraph on page 1170.
- **Restore...**: This command displays a standard open file dialog box, which can be used to designate a backup to be restored. For more information, please refer to the “Restoring Data” paragraph on page 1197.
- **Import > From File...**: This command displays a standard open file dialog box, which is used to specify a file containing data to be imported into the current table. For more information about importing data, please refer to the “Importing Data from Files” paragraph on page 1001.
- **Import > From ODBC Source...**: This command displays the system dialog box for selecting a data source, which is used to specify the ODBC source containing the data to be imported into the current table. For more information about importing data, please refer to the “Importing Data from an ODBC Source” paragraph on page 1025.
- **Export > Data to File...**: This command displays the standard export dialog box of 4D. For more information about exporting data, please refer to the “Exporting Data to Files” paragraph on page 1011.
- **Export > Data to ODBC Source...**: This command displays the system dialog box for selecting a data source, which is used to specify the ODBC source to which you want to export data from the database. For more information about exporting data, please refer to the “Exporting Data to an ODBC Source” paragraph on page 1027.
- **Export > Structure definition to XML file...**: This command displays a save document dialog box, which can be used to export the structure as an XML file. For more information about exporting the structure in XML format, please refer to the “Exporting a Structure Definition” paragraph on page 331.

- **Export > Structure definition to HTML file...**: This command displays a save document dialog box, which can be used to export the structure as an HTML file. For more information about exporting the structure in HTML format, please refer to the "Exporting a Structure Definition" paragraph on page 331.
- **Page Setup...**: The print parameters set using this command will be used for printing in all the editors during the current session. Keep in mind that the print format for forms in is set separately in the properties of each form.
- **Print...**: You can print the contents of an editor at any time. Printing will depend on the active editor.
- **Exit**: You can exit the 4D application at any time. 4D will automatically save any modifications before quitting.

Note for Mac OS Under Mac OS, the **Quit *Application*** command is placed in the Application system menu.

The Edit Menu

The **Edit** menu of 4D provides standard editing operations, find/replace commands, as well as access to the application Preferences.



Here are the menu commands provided by the **Edit** menu:

- **Undo *Operation***: Use **Undo** to cancel the last operation carried out. This menu command is useful when you make a mistake and want to start over. You can undo all the latest actions carried out in the current editor successively.

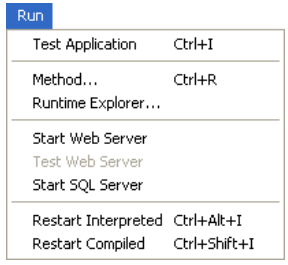
- **Redo *Operation***: Use **Redo** to repeat the last operation that was cancelled. You can redo all the latest actions that were cancelled in the current editor successively.
- **Cut, Copy, Paste**: You can select something on the screen and either cut or copy it. In either case, a copy of the selected object is placed on the Clipboard. You can then paste the object into a new location in the same window or in another window.
- **Clear**: Use **Clear** to erase a selected object. No copy is placed on the Clipboard.
- **Select All**: Use **Select All** to select every object in the current editor. For example, use **Select All** before adjusting all the elements of a form.
- **Duplicate**: You can use this command to duplicate a selected object. The duplicate is displayed slightly shifted from the original and is selected. No copy is placed in the Clipboard.
- **Find in Design**: Use this command at any time to search for a character string, an object name, etc. among all the database objects (menus, forms, methods...). A window displays the search results and allows renaming or prefixing of the objects found. For more information about searching and renaming in the database, refer to the "Searching and Replacing in the Design" paragraph on page 140.
- Commands in the **Find** > sub-menu:
  - Find commands (**Find**, **Find Next**, **Find Previous**, **Find Same**): These commands are used to carry out searches among the methods. They can only be used in the Method editor.
  - Replace commands (**Replace**, **Replace Next**, **Replace Previous**): These commands are used to carry out search-and-replace operations among the methods. They can only be used in the Method editor.
- **Show Clipboard**: You can view the current contents of the Clipboard at any time. Sometimes you will want to view the Clipboard prior to pasting its contents.

- **Preferences**: This command opens the application Preferences dialog box. The preferences determine the operation of many different aspects of the application (interface, security, backup, Web parameters, etc.) in both environments (Design and Application). Consequently, you should take care when modifying them. For more information about this, refer to the chapter “Setting Preferences” on page 161.

Note for Mac OS Under Mac OS, the **Preferences** menu command is placed in the Application system menu.

The Run Menu

The **Run** menu groups together the commands concerning the launching specific tasks in 4D as well as the running mode of the database.



It contains the following commands:

- **Test Application**: This command runs the database in Application mode without restarting (the startup code is not executed). It can be used to evaluate the functioning of the application being developed at any time.
- **Method...**: This command displays the method execution dialog box, which lets you launch a project method “manually.” For more information about using this command, refer to the “From the Execute Method Dialog Box” paragraph on page 828.

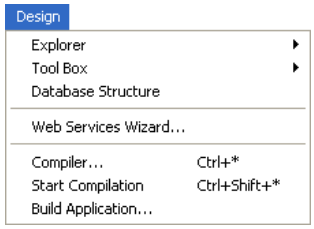
- **Runtime Explorer...**: This command displays the Runtime Explorer window. A checkmark appears next to the command when it is already displayed. The Runtime Explorer lets you check the database parameters in real time and check that the available resources are used correctly. This tool is particularly useful during the database development and analysis phase. For more information about the Runtime Explorer, refer to the section “Runtime Explorer” on page 136.

Note Only the Designer and Administrator have access to the Runtime Explorer window.

- **Start Web Server/ Stop Web Server**: These commands are active alternately. The first starts and the second stops the integrated Web server of 4D.
- **Test Web Server**: This command is available when the Web server is launched. It causes the home page of the Web site published by the 4D application to be displayed in a window of your default browser. This command lets you check the operation of the integrated Web server. For more information about the use of this command, refer to the “Testing the Web Server” paragraph on page 1164.
- **Start SQL Server / Stop SQL Server**: These commands are active alternately. The first starts and the second stops the integrated SQL server of 4D. Activating the SQL server requires a suitable license. For more information, please refer to the “Management of 4D SQL Server” paragraph on page 1165.
- **Restart Interpreted**: This command restarts the database in interpreted mode and opens it in Application mode (if at least one menu bar exists). The startup code (for example the On Startup database method) is executed. It can be used to check the final functioning of the application.
- **Restart Compiled**: This command restarts the database in compiled mode and opens it in Application mode (if at least one menu bar exists). It is enabled if the database has been compiled at least once; otherwise, it is dimmed and the database can only function in interpreted mode.
  When the database is running in compiled mode, the Design environment is not accessible. For more information about running in compiled mode, please refer to the chapter “Compilation” on page 1229.

The Design Menu

The **Design** menu gives you access to the Explorer, the Tool Box, the Structure editor, and the Web Services Wizard, as well as to the compiler and application builder of 4D:



- **Explorer** >: The sub-commands of the **Explorer** command are used to open the Explorer window to their corresponding pages. The **Explorer** sub-command at the end of the list opens the Explorer window to the default page (first or last page opened during the session). For more information about the Explorer, refer to the section “The Explorer” on page 99.
- **Tool Box** >: The sub-commands of the **Tool Box** command are used to open the Tool Box window of 4D to the page of the corresponding editor. The **Tool Box** sub-command at the end of the list opens the tool box to the default page (first or last page opened during the session). The different editors of the tool box are described in the following sections:
  - **Users**: chapter “Users and Groups” on page 831.
  - **Groups**: chapter “Users and Groups” on page 831.
  - **Menus**: chapter “Custom Menus” on page 859.
  - **Picture Library**: chapter “Picture Library” on page 891.
  - **Help Tips**: “Creating Help Tips” paragraph on page 519.
  - **Lists**: chapter “Lists” on page 903.
  - **Style Sheets**: chapter “Style Sheets” on page 923.
  - **Filters and Formats**: “Creating Custom Display Formats and Entry Filters” paragraph on page 503.
- **Database Structure**: This command displays the Structure editor window of 4D, which you can use to define the tables, fields and relations of your database. For more information about this editor, refer to the “Using the Structure Editor” paragraph on page 243.

- **Web Services Wizard...**: The **Web Services Wizard** command displays a wizard that can be used to facilitate the integration of Web Services into your 4D database. For more information, refer to the "Subscribing to a Web Service in 4D" paragraph on page 1149.
- **Compiler...**: The **Compiler** command displays the window for the built-in compiler of 4D. For more information, refer to the "Compiler Window" paragraph on page 1236.
- **Start Compilation**: The **Start Compilation** command directly executes the compilation of the database with the current parameters, without passing by the Compiler dialog box. For more information, please refer to the "Compiler Window" paragraph on page 1236.
- **Build Application...**: This command is used to access the Build Application dialog box. It can only be accessed if the database has been compiled. For more information, refer to the chapter "Finalizing and Deployment of 4D Applications" on page 1251.

The Records Menu

The commands of the **Records** menu are used for entering, modifying, selecting, searching and sorting data displayed in the records window (see the "Record Display Window" paragraph on page 79).

These commands are the following:

- **Show Current Table (*Table_Name*)**: This command displays the records of the current table in the current output form (the name of the current table is indicated in parentheses). If there is no current table, the command is dimmed.

- **Last Used Tables** > (***Table Names***): This submenu lists the last 15 tables used (that is, displayed) in the database. Selecting one of these table names causes the records of this table to be displayed in the current output form in the records window.
- **List of Tables**: This command displays the List of tables window, which can be used to browse among the tables and forms of the database. This window is described in the "Browsing Different Tables and Forms" paragraph on page 947.
- **New Record in List**: This command adds a blank line in the current output form and places the cursor in the first field to begin data entry.
- **New Record**: This command displays a blank input form and places the cursor in the first field to begin data entry.
- **Modify Record**: This command displays the selected (highlighted) record using the current input form. **Modify Record** is dimmed if no records are highlighted or if you are currently using an input form.
- **Delete Selection**: This command deletes the highlighted records from the data file (equivalent to the **Clear** command of the **Edit** menu).
- **Show All**: This command displays all the records of the current table in the current output form. **Show All** is dimmed if you are currently using an input form.

Note For more information on entering, modifying and updating records, please refer to the "Editing Records" paragraph on page 972.

- **Show Subset**: This command displays only the group of records that is highlighted in the current output form. **Show Subset** is dimmed if no records are highlighted or if you are using an input form.
- **Query** > **Query**: This command displays the standard query editor of 4D. This editor is a general purpose search editor that you can use to perform many types of queries.
- **Query** > **Query by Example**: This command displays the current input form as a query-by-example window. The search results are displayed in the current output form.
- **Query** > **Query and Modify**: This command also displays the current input form as a query-by-example window. When you use Query and Modify, the first record located by the query is displayed in the current input form.

- **Query** > **Query by Formula**: This command displays the Formula editor. Use this command when you want to search for records based on the results of a formula.

Note For more information about selecting and searching for records, refer to the "Selecting and Searching Records" paragraph on page 950.

- **Order By**: This command displays the Order By editor. This editor allows you to sort the records of the current selection.

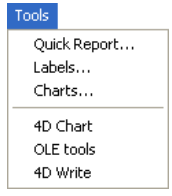
Note For more information about sorting records, refer to the "Sorting Records" paragraph on page 987.

- **Apply Formula**: This command displays the Formula editor. Choose **Apply Formula** to make changes to the current selection of records using a formula.

Note For more information about the Formula editor, please refer to the "Global Updates" paragraph on page 979.

The Tools Menu

The **Tools** menu contains the commands that call the Quick Report, Label and Chart editors of 4D and lists the plug-ins installed in your database:



- **Quick Report**: This command displays the Quick Report editor. Use it to create and print columnar reports.
- **Labels**: This command displays the Label editor. Use it to create and print labels.
- **Charts**: This command displays the Chart Wizard. Use it to graph data.

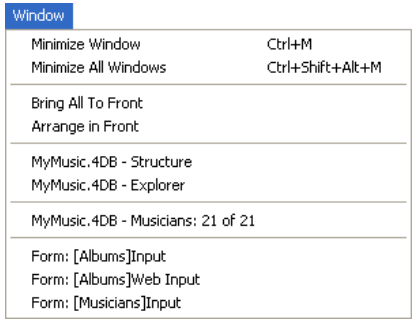
Note These editors are described respectively in chapters "Quick Reports," page 699, "Labels," page 1031 and "Graphs," page 1053.

The **Tools** menu also lists the different plug-ins installed in your database. When you select a plug-in from this menu, a new external window containing this plug-in is created.

By default, this menu contains the integrated **4D Chart** plug-in command, which is used to create and manipulate graphs, as well as the **OLE tools** command (under Windows only), which is used to insert OLE objects into your databases.

The Window Menu

The **Window** menu is used to manage the different windows open in the application. The first four commands are standard:



- **Minimize Window**: Minimizes the foreground window.
- **Minimize All Windows**: Minimizes all the windows open.
- **Bring All To Front**: Enlarges all the minimized windows.
- **Arrange in Front**: Displays all the windows in a cascade.

  All the open windows are then listed in the menu. A check mark (√) indicates the current window. To bring a window to the foreground, you simply need to select it using this menu.

  To close all the open windows, hold down the **Alt** key (under Windows) or the **Option** key (under Mac OS), when you click on the close box of the current window.

The Help Menu

The **Help** menu can be used to access help, security and maintenance functions of the application:

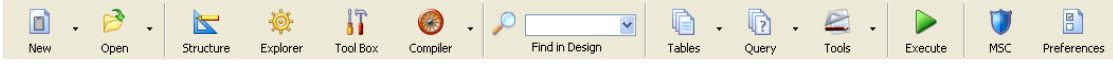
- **4D Help**: Access to the on-line help of the application.

- **Maintenance Security Center**: This command displays the Maintenance and Security Center, which groups together all the tools required for checking, maintenance, backup and compacting of the data and structure files of your applications. For more information, please refer to the chapter “Maintenance and Security Center” on page 1209.
- **Update License...:** This command can be used to add license numbers. For more information about this function, please refer to the 4D *Installation Guide*.

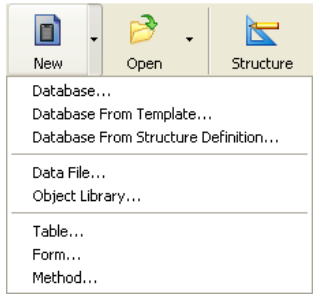
  Under Windows, the **Help** menu also contains the **About 4D Developer...** command; under Mac OS, this command is found in the Application menu. The About dialog box provides various information about the 4D application being run, the copyrights used and the company 4D, as well as a page of links to various on-line information sources.

The Toolbar

By default, in 4D a toolbar is displayed just below the standard menu bar. This toolbar gives you direct access to the main functions of the application.



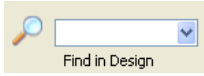
- The buttons with a triangle icon to their right have associated menus offering different possible actions:



The **New** button works as follows: clicking on the button executes the current action associated with it. This action is the last one selected in the menu. To display the menu and select a different action, click on triangle icon in the right-hand edge of the icon.

The other buttons of this type execute the first action of the associated menu when clicked directly.

- The Find in Design area allows launching a standard database search. To launch a search directly, type the name of the string to find in the entry area then hit **Return**.



The search is done without options (i.e., of the "contains" type in all the database objects).
If you want to perform a search with options, you can click on the magnifying glass icon to display a standard search dialog box. Refer to the "Searching and Replacing in the Design" paragraph on page 140.

If you don't want to display the toolbar in the Design environment, you can hide it by deselecting the corresponding option in the "Display Toolbar" area on the **Application** > **Options** page of the application Preferences. For more information, refer to the "Options Page" paragraph on page 163.

When you create custom menus, you can create custom toolbars and add buttons corresponding to your custom menu items. For information on creating custom toolbars, see the "Toolbar Icons" paragraph on page 885.

Custom Keyboard Shortcuts

You can customize the default keyboard shortcuts of the Design environment in order to adapt them to your working habits. All the shortcuts of the Design environment are referenced in an XML file named **4DShortcuts.xml**, located in the **4D Extensions** folder. To specify custom shortcuts, you simply need to duplicate this file and modify the copy using an XML editor (make sure not to assign the same shortcut to more than one action). Once the modifications are finished, place the copy in the current Preferences folder of 4D. The **4DShortcuts.xml** file found in the Preference folder will then be loaded by 4D instead of the default file.

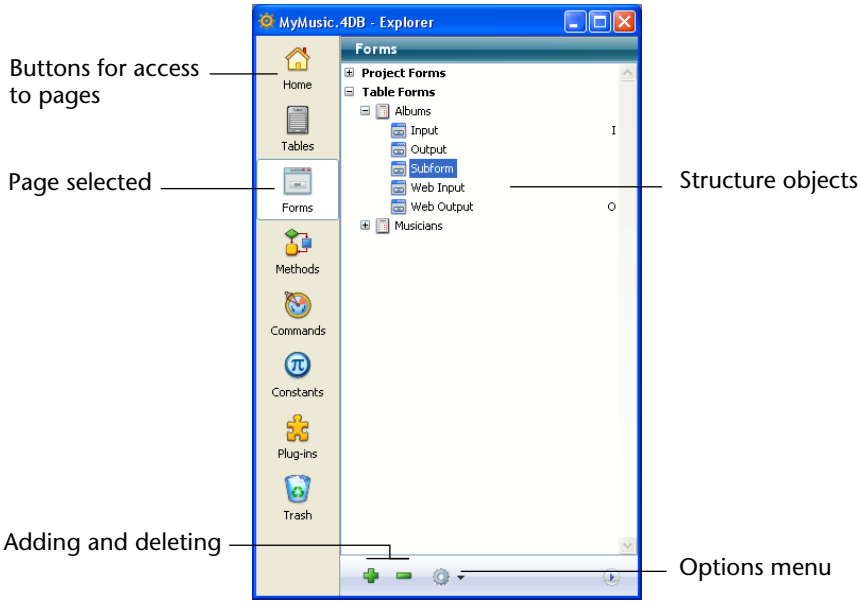
WARNING: *Do not modify the 4DShortcuts.xml file directly in the 4D Extensions folder.*

The Explorer

The Explorer is a window in the Design environment that gives you convenient access to tables, forms, methods, built-in 4D commands, constants and plug-ins. You can display the Explorer at any time by choosing one of the pages in the **Design** > **Explorer** sub-menu or by clicking on the Explorer button in the toolbar.

Working with the Explorer

The buttons on the left side of the Explorer let you access the different pages for Design environment objects. The Explorer has separate pages for tables, forms, methods, commands, constants and plug-ins.
The Home and Trash pages provide additional specific functions.
When you display a specific page, the corresponding objects are listed in the Explorer. On each page, the objects are displayed in hierarchical lists. A control area below the list contains adding and deleting buttons as well as a menu of options;



Renaming an Object

If you need to rename a folder, a table, a field, a form or a method, hold down the **Alt** key (under Windows) or the **Option** key (under Mac OS) and click the name of the object. You can also click twice on the name (but wait a moment between the two clicks, otherwise it becomes a double-click). The text then becomes editable.

Make your changes and then click anywhere outside the text area to save your changes.

Note Changing the name of a form invalidates the methods and formulas which use its previous name. Each of these objects must be updated in order to operate correctly.

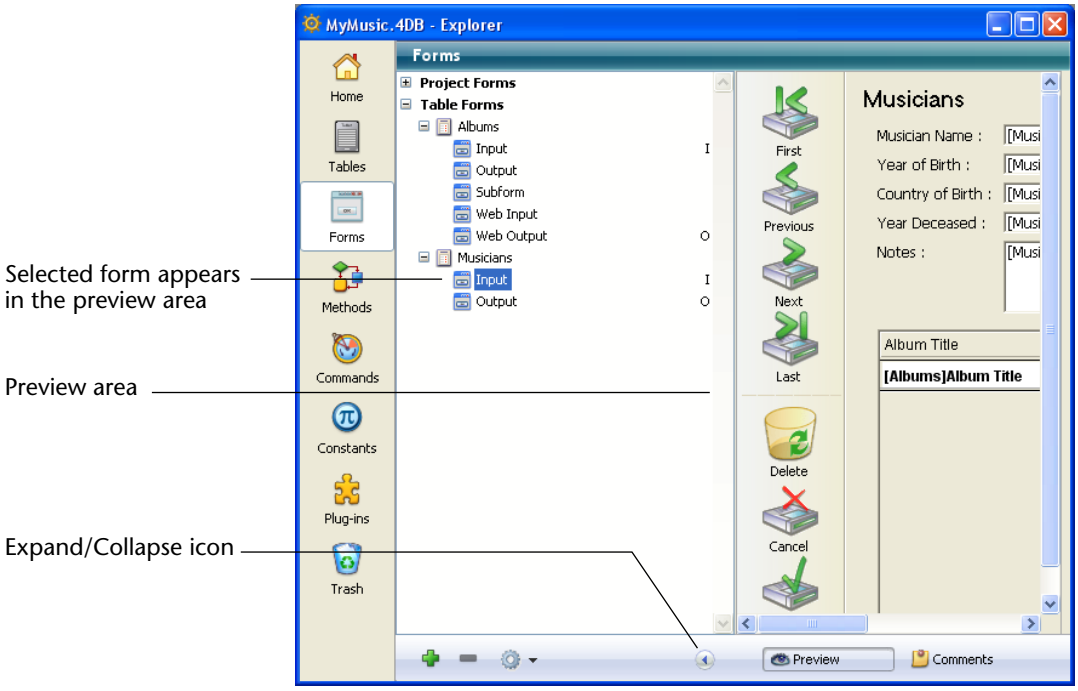
The Explorer always lists objects alphabetically. If the new name changes the sort order, 4D will resort the list when you click outside the entry area.

4D Server The object name is modified on the server when you click outside the entry area of the name. If several users modify an object name at the same time, the final name will be the one given by the last user who modified it.
You can set an owner for an object so that only certain users can change its name. For more information about setting access rights, refer to the chapter “Users and Groups” on page 831.

Displaying and Hiding the Preview Area

You can display or hide the Explorer preview area by clicking on the expand/collapse icon for the preview area  . The preview window lets you preview table images, forms, methods, command documentation, constant values, as well as information about the components and plug-ins installed in the database. You also can use the preview area to enter and view comments about certain database objects.

The following illustration shows a form being previewed:



To hide the preview area, click the expand/collapse icon again.

Resizing the Explorer Window

You can resize the Explorer by dragging the lower-right corner of the window.

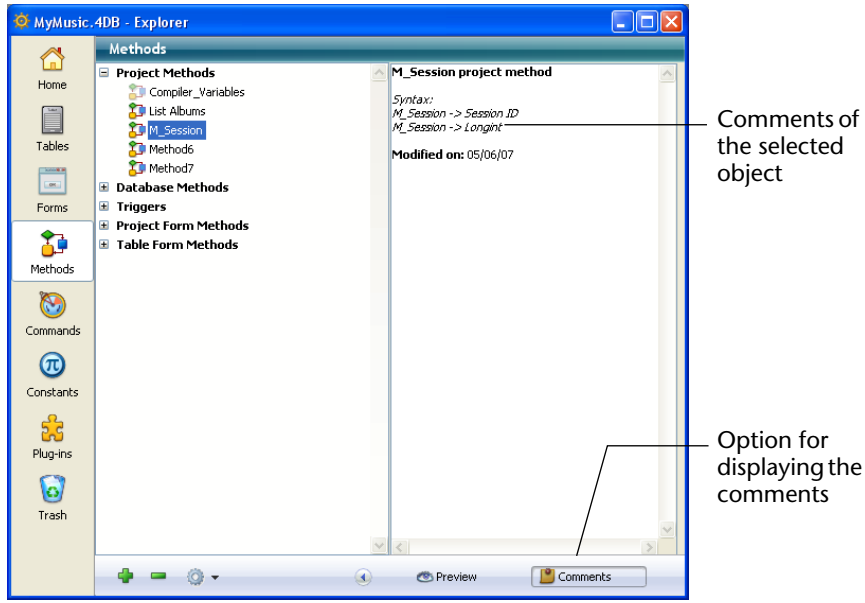
Note If the preview area is not displayed, you cannot display it by resizing the window. You must use the icon.

If the preview area is not displayed, you can only expand the Explorer window vertically. When the preview area is displayed, you can expand the Explorer window both horizontally and vertically.

Displaying Comments

4D allows you to assign comments to the following database objects: folders, tables, fields, forms and methods (database methods, project methods, triggers and form methods) of your database. Displaying and modifying comments are carried out using the Explorer (the preview area must be expanded).

To display and/or modify an object's comments, select the object in the object list and click the **Comments** button located below the preview area. The preview area is then replaced by the comments area.



The creation and use of comments are described in detail in the section "Using Comments" on page 132.

Using Drag and Drop

In many instances, you can use drag and drop to add a database object to an editor window. For example, you can add a field to a form by dragging a field name from the Tables page of the Explorer to an open form in the Form editor. When you are working with the Method editor, you can add the names of tables, forms, fields, project methods, constants, and commands, as well as their syntax, to a method using drag and drop.

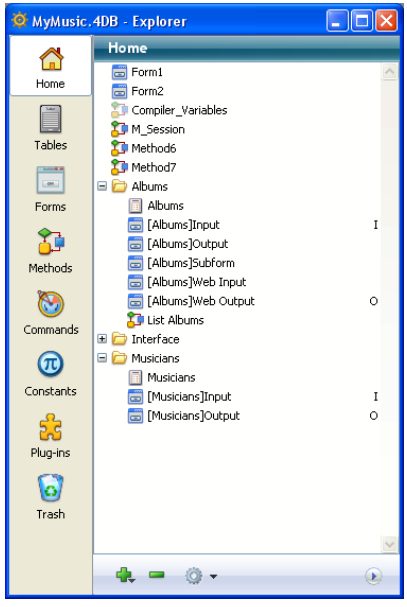
The sections that describe each Explorer page give specific information on the drag-and-drop options for that page. Each Explorer page is described briefly in the following sections. More detailed information is presented in the chapter that deals with the appropriate topic.

Using the Contextual Menus and the Options Menu

You can use contextual menus and the Options menu on the Home, Forms, Methods and Trash pages of the Explorer. These menus provide access to additional functions. They are described in the following pages.

Home Page

The Home page lets you set up and use “folders” of objects in the 4D Explorer.



In keeping with folders used in the Windows and Mac OS operating systems, 4D folders group together different objects according to custom criteria (in particular by function) and not only by type. For example, you can group objects related to handling e-mails in your database in a folder named “E-mail.”

Table, Form and Method objects can be placed in folders. It is also possible to create sub-folders on several different levels.

All customized database objects (tables, forms, project methods) are always displayed on the page. Note that they remain listed on the other pages of the Explorer as well.

On this page, you can add folders or subfolders, rename, move and reorganize folders and objects. You can also create tables, forms and project methods directly. As on the other pages of the Explorer, you can open a table, a form or a project method in a window of the corresponding editor by double-clicking its name, or insert it using drag-and-drop. Finally, you can rename or delete folders and objects.

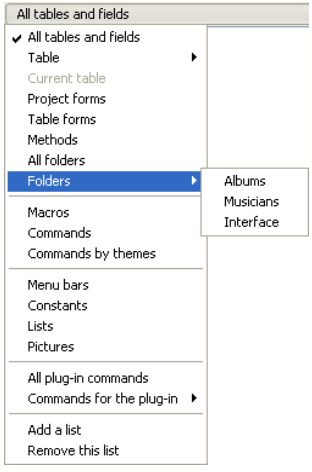
It is also possible to assign a folder to each object when it is created.

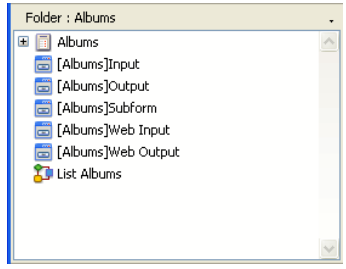
4D Server All client machines connected to the same database share the same folder configuration. Once a change is made by a client machine on the Home page (move, addition, etc.), it is automatically and instantaneously made on all connected client machines.

Why Use Folders?

Folders are useful for organizing databases by functionality or customized criteria (dates, languages, etc.). Accessing different objects from the Explorer is thus made easier.

Folders are also accessible in the form of hierarchical lists in the 4D Method editor. This makes calling appropriate objects easier when writing methods:



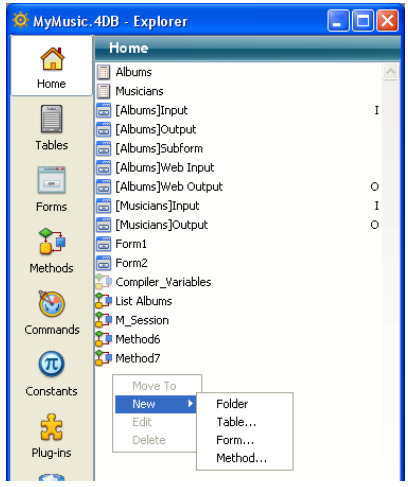


It is also possible to use folders to filter the display of the tables in the Structure editor (see the "Highlight/Dim Tables by Folder" paragraph on page 252).

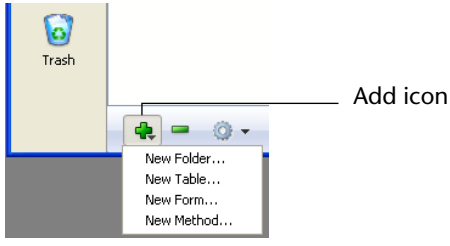
Creating Folders or Subfolders

By default, no folder is created and all the objects are placed at the top level.
You can add folders or subfolders at any moment in the Explorer **Home** page.

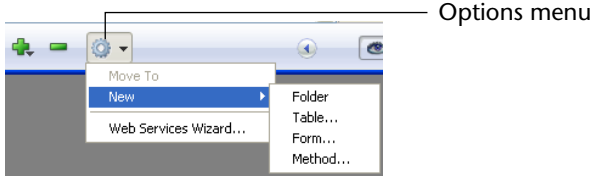
To do so, simply use the **New>Folder** command from the contextual menu of the area (**right-click**):



You can also use the menu that appears when you click the add icon located at the bottom of the area:



And finally, the options menu of the page can be used to create objects:

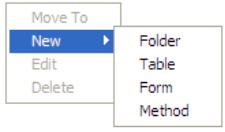


The folder or subfolder is immediately created and its default name is "*Folder_+folder number.*" You can modify this name as desired; however, the name of the folder or subfolder must be unique and no longer than 31 characters.

- To create a folder, make sure that no item is selected when you add the new folder.
- To create a subfolder, select the folder in which you wish to add a subfolder before creating the new folder.

Afterwards, you can easily change a folder to a subfolder and vice-versa using drag-and-drop or move commands.

The **New>** command of the Home page allows creating not only folders, but tables, forms and project methods as well.

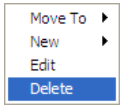


When you choose this type of subcommand, you go directly to the dialog box used for creating the desired objects.

Note The contextual menu and the options menu of the Home page provide additional functions depending on the type of object selected. The options available per type of object are described in the following pages.

Deleting an Object

The **Delete** command causes the deletion of the selection. If the selection contains a folder or subfolder, all the items contained in them are deleted:



This command can be used with any type of item (folders, subfolders, tables, forms and project methods) and with multiple selections. Deleted objects disappear from lists and application editors.

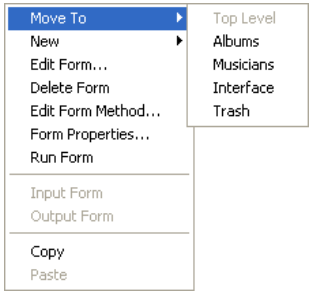
Deleted items are placed in the **Trash** of the Explorer. For more information on how the Explorer Trash works, please refer to the "Trash Page" paragraph on page 129.

To delete one or more objects from the Explorer **Home** page, it is also possible to select them then choose the **Move To>Trash** command in the contextual menu (**right-click**) or using the options menu of the page.

Move To

The **Move To** command of the contextual menu on the **Home** page allows modifying the contents of a folder without having to expand the destination folder beforehand.

To do this, **right-click** with the mouse on the object, folder or selection. The contextual menu is displayed. The hierarchical **Move To** command provides all the existing folders and subfolders as potential destinations:

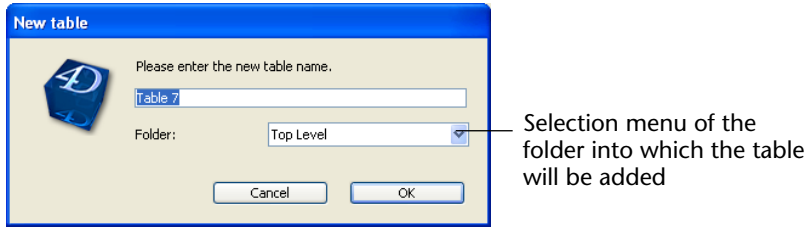


Simply select the folder into which the selection will be placed. If you choose **Top Level**, the selection is placed at the first level of the list, outside any folder (this item is dimmed when trying to move an object that is already at the first level).
If you choose **Trash**, the selection is deleted (see the "Deleting an Object" paragraph on page 106).

Adding an Object to a Folder During Creation

You can place a new object directly in a folder at the time of its creation. The "Folder" menu, located in the dialog box for creating tables, forms and project methods, allows setting the folder into which the object should be created. By default, objects are created at the first level (Top Level) or in the last folder selected in the Explorer (if applicable):

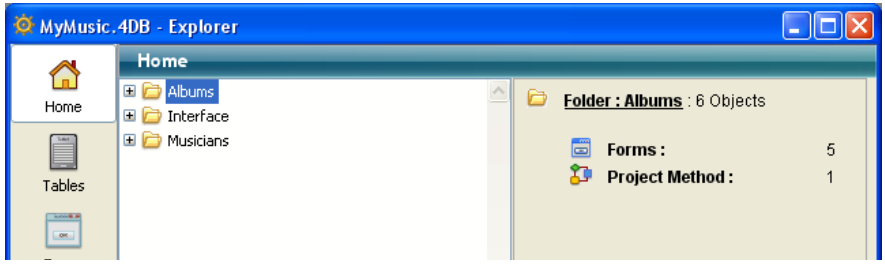
Example: Table creation dialog box



Of course, you can always change these folder objects once created on the Explorer **Home** page.

Folder Information

When a folder is selected in the Explorer, the preview area of the window displays several pieces of information:



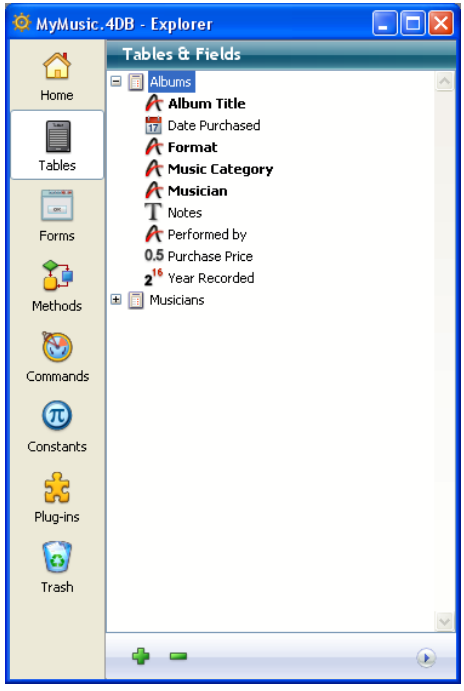
- **Folder: Folder name**: Number of objects available inside the folder.
- **Table(s)**: Number of tables contained in the folder.
- **Form(s)**: Number of forms contained in the folder.
- **Project Method(s)**: Number of project methods contained in the folder.
- **Folder(s)**: Number of folders (subfolders) contained in the folder.

If an object type is not present in a folder, it does not appear in the preview area. If a folder is empty, the area only displays **Folder: Folder name**.

When several folders and/or objects are selected, the information displayed concerns the current item, in other words, the last item on which you clicked.

Tables Page

The Tables page lists all the tables and fields in the database. It can be used as an alternative to the Structure editor to access table and field properties. When a table is expanded, the fields in the table are shown.



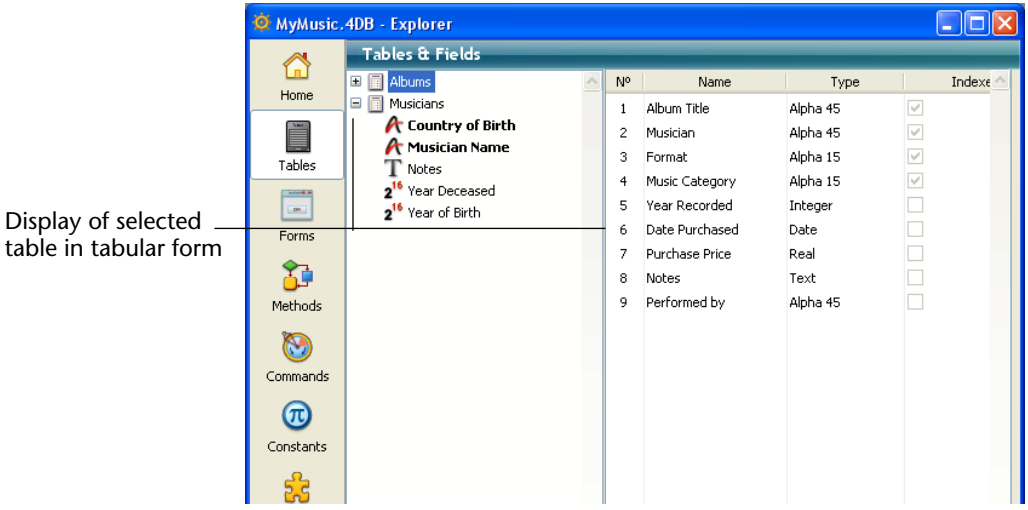
The field type is indicated by an icon to the left of its name. Double-click a field name to display its properties in the Inspector palette. For more information on field properties, see “Creating Fields and Setting Field Properties” on page 265.

Using Drag and Drop

You can add a field to a form by dragging the field name from the Tables page onto the form. You can add either a table name or a field name to a method by dragging the name to the method. When you do so, the name appears using the correct syntax. For example, if you drag the field “First Name” in a [Customers] table, it appears in the Method editor as “[Customers]First Name.”

Preview Area

You can also preview the table in tabular form in the preview area of the Explorer. To do so, select a table and click the Preview icon (if the preview area is not already displayed). The following illustration shows a table being previewed:

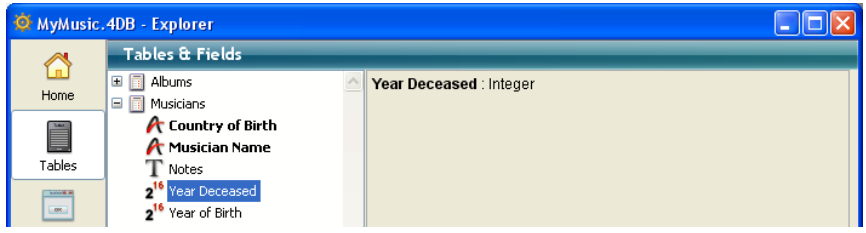


Display of selected table in tabular form

This area is for information only; it is not possible to modify the values.

In addition to the numbers, names and types of fields for the selected table, the indexes associated with the table are also listed (regardless of their type) as well as any relations originating from the table. For each field at the origin of a relation (Many field), the destination field (One field) is indicated in the Relation column.

You can also view the field type when you highlight its name in the list:



Viewing a Table Image in the Structure Window

You can bring a table image in the Structure editor window into view by double-clicking the table name. When you do so, 4D centers the table image you clicked in the Structure editor window and displays its properties in the Inspector palette. This feature is useful if you have a large structure and would otherwise need to scroll the Structure editor window to view a particular table image.

Adding a Table or Field

You can add a table or field using the add button located at the bottom of the page.

To create a table, select a table name (or make sure that no object is selected) and click on the add button. The standard New table dialog box appears (refer to the "Creating and Modifying a Table" paragraph on page 256).

To create a field, select an existing field and click on the add button. The Inspector palette appears, ready to create a field in the selected table (refer to the "Creating a Field" paragraph on page 265).

Deleting a Table

You can delete a table using the delete button . When you select a table and click on this button, the table is deleted from the 4D editors and can no longer be used. The forms and methods associated with it are also deleted.

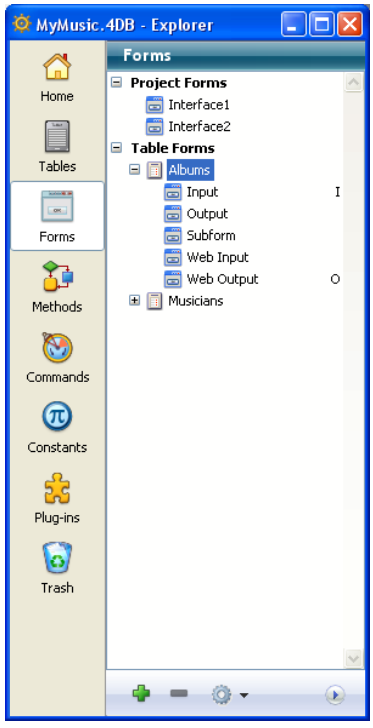
Note It is not possible to delete a field using the delete button.

In reality, the table is not permanently deleted, but rather placed in the Trash and hidden in 4D (see the "Trash Page" paragraph on page 129). Its actual deletion will only take place when the Trash is emptied.

Note It is possible to permanently delete a table from the Structure editor (see the "Deleting a Table" paragraph on page 263) or via the SQL language.

Forms Page

The Forms page contains two lists of forms: **Project Forms** and **Table Forms**. The list of table forms displays the tables of the database and their associated forms:



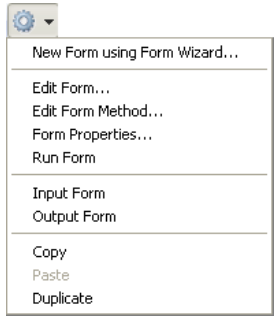
The **project forms** are independent forms that are not attached to any table. These forms are particularly useful when developing interfaces and components.

The **table forms** are forms that are associated with a specific table. They benefit from automatic functioning and additional functions for processing data. For more information about the differences between these two types of forms, please refer to the "Table Forms and Project Forms" paragraph on page 339.

Creating a New Form

The Forms page can be used to create a form (project form or table form) in two ways:

- Using the Form Wizard: To do this, select either the title “Project Forms” (to create a project form) or the table (or the form of a table) in which you want to add a new form (to create a table form) and then choose **New Form using Form Wizard...** in the options menu of the page:



  The Form Wizard appears, ready to create a new form. For more information about using this wizard, refer to the “Creating a New Form using the Wizard” paragraph on page 350.

- By creating a blank form: To do this, select either the title “Project Forms” (to create a project form) or the table (or the form of a table) in which you want to add a new form (to create a table form) and then click on the add button at the bottom of the page  . A dialog box appears, which lets you set the name and folder of the form. When you validate this dialog box, the blank form is created and displayed in the Form editor window. For more information, refer to the “Creating a Blank Form Using the Explorer” paragraph on page 367.

Editing a Form

From the Explorer, you can open an existing form directly in the Form editor for modification. To do this, you can:

- Double-click on its name,
- Select the form and then choose **Edit Form...** in the contextual menu or in the options menu of the page.

  For more information about editing forms, refer to the chapter “Form Editor” on page 371.

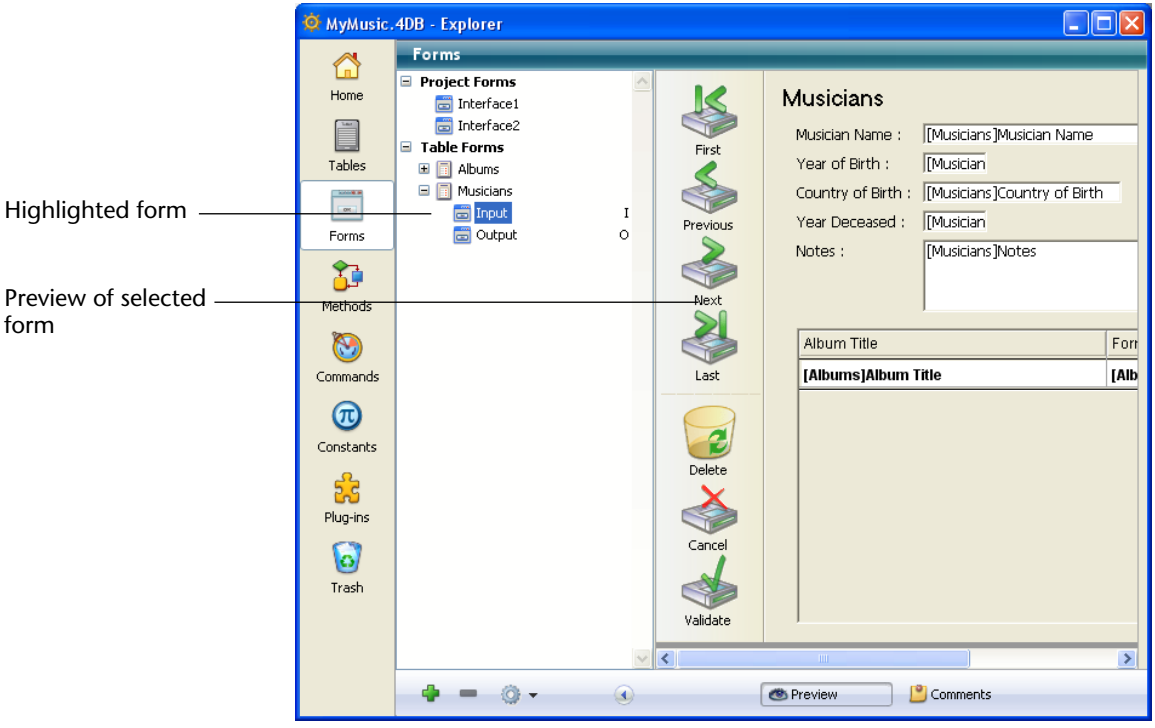
Deleting a Form

To delete a form, select its name in the list and click on the delete button or use the **Delete Form** command of the contextual menu. You can also drag and drop the form into the Trash.

Note You cannot delete a form if it is the current default input or output form for the table.

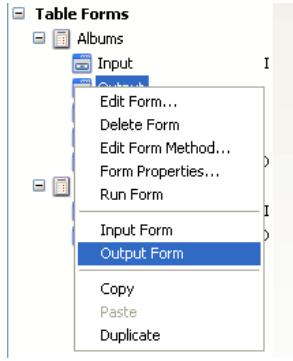
Previewing a Form

To display the preview area, click the icon. Highlight a Form in order to preview it.



Designating the Current Input or Output Form

You can designate the current input and output forms for each table using the Explorer (table forms only). To do this, click the desired form name in the hierarchical list and then use the **Input Form** or **Output Form** commands of the contextual menu:



For more information on default input and output forms, see "Designating the Input and Output Forms" on page 368.

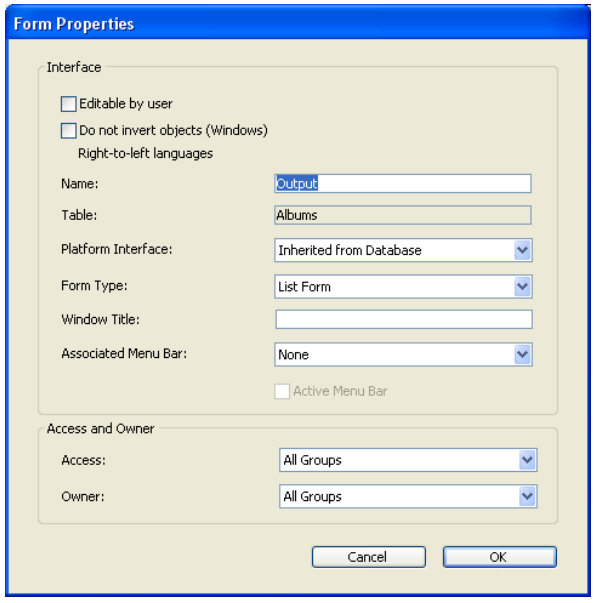
Modifying the Form Method

You can open a form method directly from the Forms page of the Explorer by choosing the **Edit Form Method...** command from the contextual menu or from the options menu. The form method will be opened in the Method editor window.

Setting Form Properties via the Explorer

You can access the main form properties from the Explorer. To do so, click on the name of the desired form, then choose the **Form Properties...** command in the contextual menu or in the options menu of the page.

The properties are displayed in a specific dialog box:



For more information about the properties found in this dialog box, refer to the "Setting Form Properties" paragraph on page 394.

Executing a Form

You can execute a form in its context (list of records for a list table form and current record page for a detail table form) in the records display window. To do so, choose **Run Form** in the contextual menu or from the options menu of the page. For more information about testing forms, please refer to the "Form Editor Toolbar" paragraph on page 377.

Using Drag and Drop

You can add a form name to a method by dragging. When you do so, the form name appears using the correct syntax. For example, if you drag the form "Input" in the [Company] table, it will appear in a method as [Company];"Input".

You can add a subform to another form by dragging the name of the List form from the Forms page of the Explorer to the open form in the Form editor. You can add a Detail subform by holding down the **Shift** key and dragging the name of an input form from the Forms page of the Explorer to the subform area on the form. For more information, see the section "Adding a Subform to the Form" on page 621.

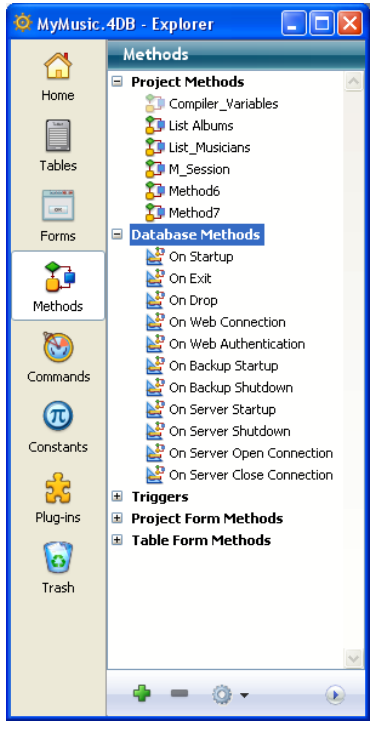
Copying, Pasting or Duplicating a Form

4D lets you copy/paste or duplicate existing forms via the corresponding commands in the contextual menu or in the options menu of the page. This makes it quick and easy to develop several different forms having the same characteristics .

4D automatically adds and increments a number to the form name if a form with the same name already exists at the target location.

Methods Page

The Methods page lists the project methods, database methods, and triggers, as well as project and table form methods for the database. These types of methods are grouped by category.



Note If components with shared methods have been installed in the database, the page displays an additional category: **Component Methods**. This item lists the components installed and their shared methods. This point is detailed in the "Display of Components" paragraph on page 1278.

Creating a New Method This paragraph describes how to create each type of method from the Explorer.

- **Project Method**s: To create a new project method, highlight the Project Methods item in the hierarchical list, or the name of an existing project method, and click the add button .

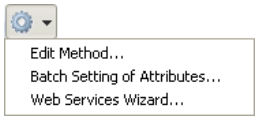
Note You can also create a project method by duplication or copy-paste (see the "Copying, Pasting or Duplicating Project Methods" paragraph on page 122).

- **Database Methods**: You cannot create new database methods. Instead, you can add code to an existing blank database methods. To do so, expand the Database Methods item then double-click the database method to be modified or choose **Edit Method...** from the contextual menu or from the options menu of the page.
- **Triggers**: To create a trigger, expand the Triggers item in the hierarchical list; highlight the desired table; then click on the add button  or double-click on the table name. You can also choose **Edit Method...** from the contextual menu or from the options menu of the page.
- **Project Form Methods**: To create a new method for a project form, expand the Project Form Methods item in the hierarchical list; expand the desired table, and then highlight the desired form. Click the add button  or double-click on the form name. You can also choose **Edit Method...** from the contextual menu or from the options menu of the page.
- **Table Form Methods**: To create a new method for a table form, expand the Table Form Methods item in the hierarchical list; expand the desired table, and then highlight the desired form. Click the add button  or double-click on the form name. You can also choose **Edit Method...** from the contextual menu or from the options menu of the page.

Note You can also create a new form method by selecting **Edit Form Method...** in the contextual menu or the options menu of the Forms page.

Creating a Method with the Web Services Wizard

You can display the Web Services Wizard from the Methods page. To do so, select **Web Services Wizard**... in the options menu of the page.



The Web Services Wizard appears, ready to be used to discover a Web Service and generate the corresponding proxy methods. For more information about using this wizard, refer to the "Subscribing to a Web Service in 4D" paragraph on page 1149.

Deleting Methods

You can delete any method by highlighting it and clicking on the delete button at the bottom of the page, or by choosing **Delete Method...** from the contextual menu.

The manner of deletion depends on the method type:

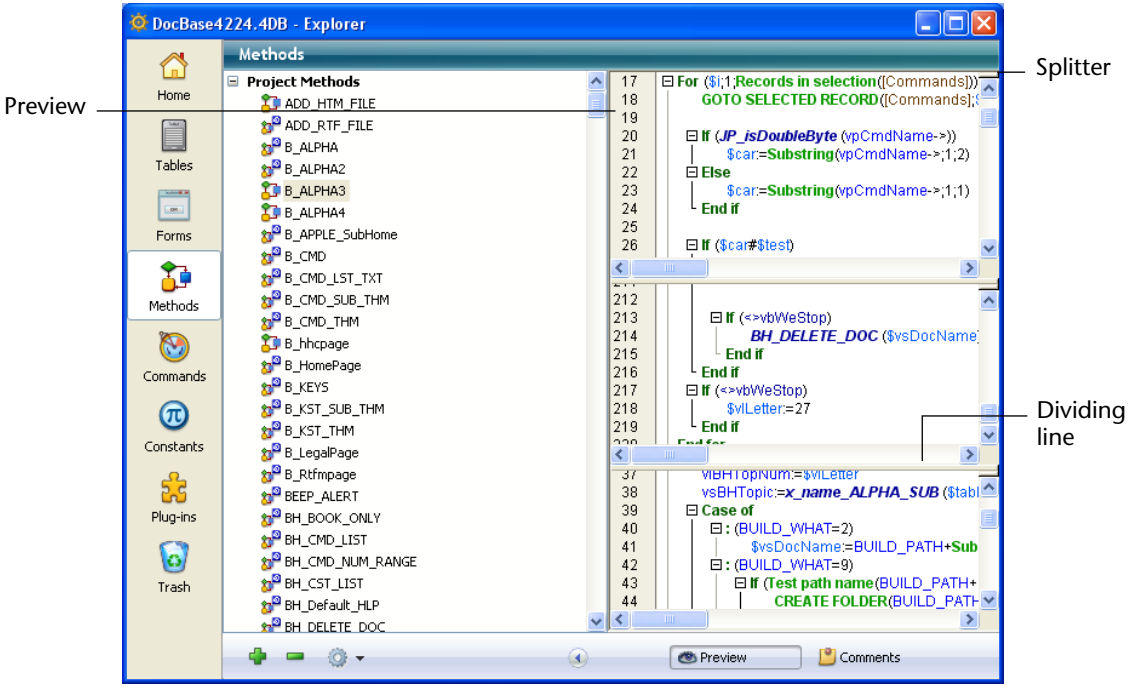
- Database methods, Triggers and Form methods: The contents of the method are erased and the method returns to its default "blank" state. A warning dialog box indicates that the deletion is permanent.
- Project methods: The method is deleted from the list of methods and placed in the Trash. It is possible to recover it from the Trash or to delete it permanently (for more information, refer to the "Trash Page" paragraph on page 129).

Modifying a Method

To modify an existing method, double-click its name, or choose **Edit Method...** in the contextual menu or in the options menu of the page. The method is then opened in the Method editor.

Previewing a Method

Click the icon to display the preview area and highlight the method you want to preview. The preview area can be divided into several panes containing scroll bars, like the windows of the Method editor:



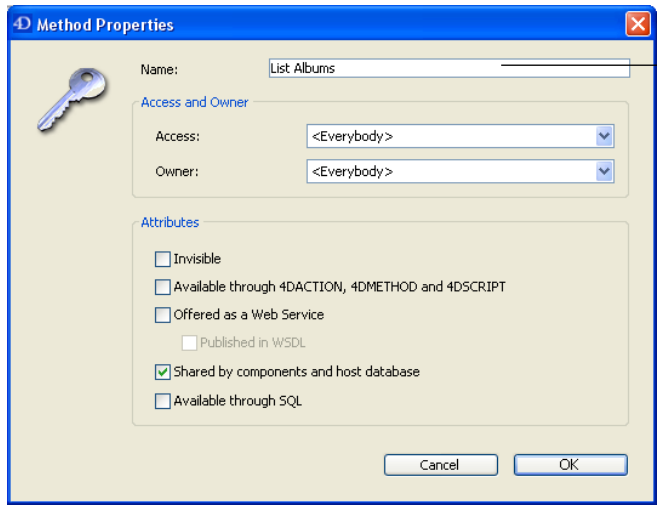
To add an area, click on the splitter and drag it downwards. To remove an area, double-click on its dividing line.

You can highlight, then copy or drag and drop, all or part of the contents of the different areas.

Project Method Properties

You can display or modify the properties of project methods from the Explorer. To do so, highlight a method and choose the **Method Properties...** command in the contextual menu or the options menu of the page.

The “Method Properties” dialog box appears, allowing you to modify the name as well as other method properties.



The icons associated with the methods vary according to the type and properties of the method. For instance, the following is the icon of a project method offered as a Web Service:  .

For more information about the other options of this dialog box, refer to the “Defining the Properties of Project Methods” paragraph on page 773.

Batch Setting of Attributes

The **Batch setting of attributes...** command of the contextual menu or options menu (applied to project methods only) is used to modify an attribute (Invisible, Available through 4DACTION, 4DMETHOD and 4DSCRIPT, etc.) for all or part of the database project methods in a single operation.

For more information about this command, refer to the “Batch Setting for Method Attributes” paragraph on page 777.

Executing a Method

It is possible launch the execution of a database method and/or a project method from the Explorer. To do this, select the method then choose **Run Method...** from the contextual menu or the options menu of the page.

The method is executed in standard mode. For more information, please refer to “Execution of a Method” paragraph on page 762.

Using Drag and Drop

When you are writing a method, you can add the name of another project method using drag and drop. Highlight the desired method name in the Explorer and drag it to the Method editor window.

When creating custom menus, you can associate a project method with each menu command. You can do so by dragging a project method name from the Explorer to the Menu Bar editor. For more information, see the "Associating a Project Method" paragraph on page 876.

Copying, Pasting or Duplicating Project Methods

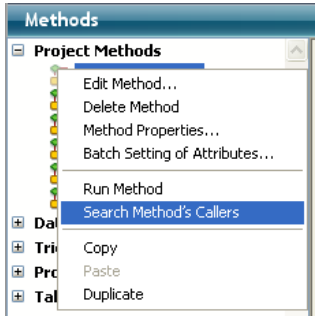
It is possible to copy, paste and duplicate project methods from the methods list of the Explorer. These functions allow a method to be duplicated in the same database or to be copied from one database to another.

To duplicate a method, select it and choose **Duplicate** in the contextual menu or in the options menu of the page. To copy a method, select it and choose **Copy** in the contextual menu. To paste the method, choose **Paste** in the same contextual menu.

In the case of duplication, or when a method with the same name already exists in the place where you paste the method, a number is added to the method name, for example *MyMethod1*. This number is incremented when necessary (*MyMethod2, MyMethod3*, etc.).

Searching for Callers

From the Explorer, you can search for a list of objects that reference a specific project method (other methods or menus). This function is available via the **Search Method's Callers** command in the contextual menu or the options menu of the page:

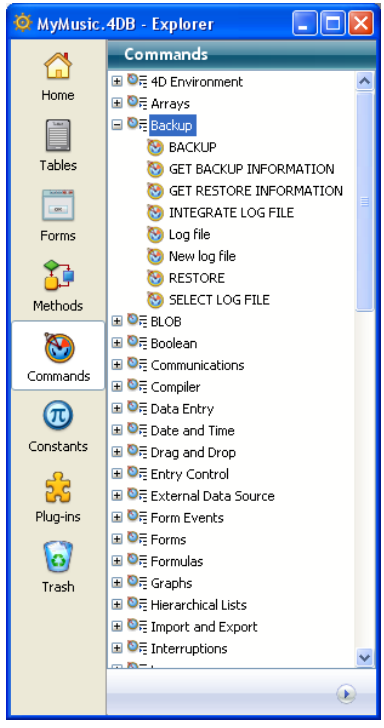


Note The **Search Callers** command is also available in the contextual menu of the Method editor.

This command displays a new results window (the objects found are displayed in the Callers list). For more information about the search results window, please refer to the “Results Window” paragraph on page 146.

Commands Page

The Commands page displays all built-in 4D commands, grouped by category. It is equivalent to the list of commands shown in the Method editor.

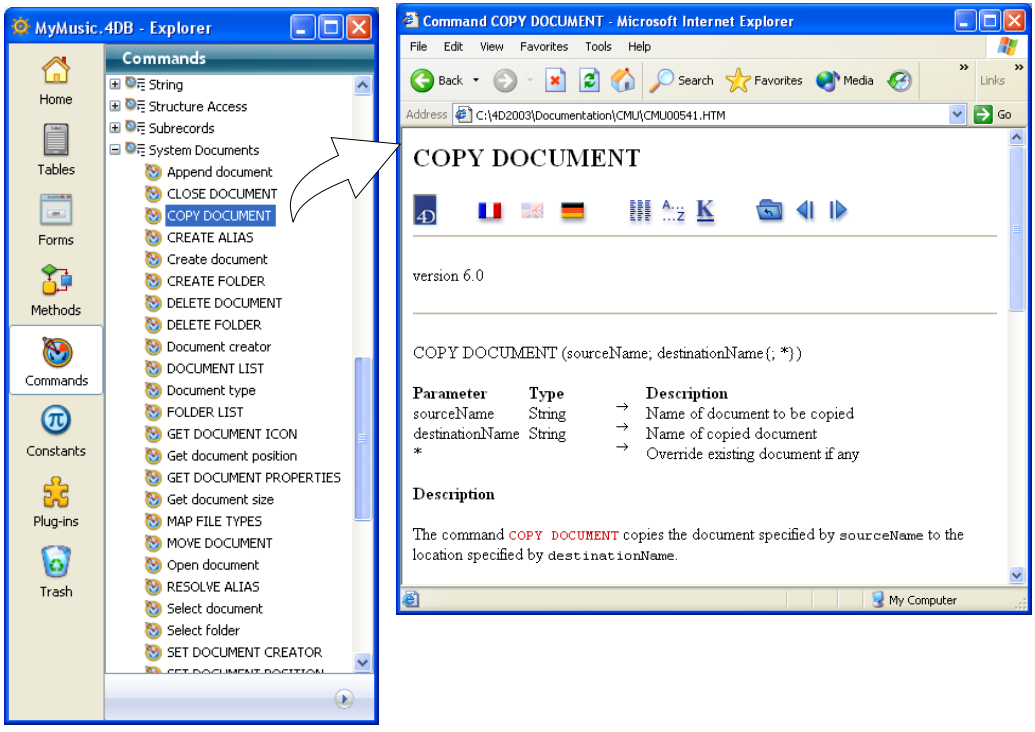


You can use this page to access on-line documentation for the commands. The preview area also allows you to display either the documentation (only under Mac OS) or information about the command syntax (Windows).

Access to the On-line Documentation

You can access the on-line (HTML) documentation of 4D directly from the Commands page: to do this, double-click on the name of the command.

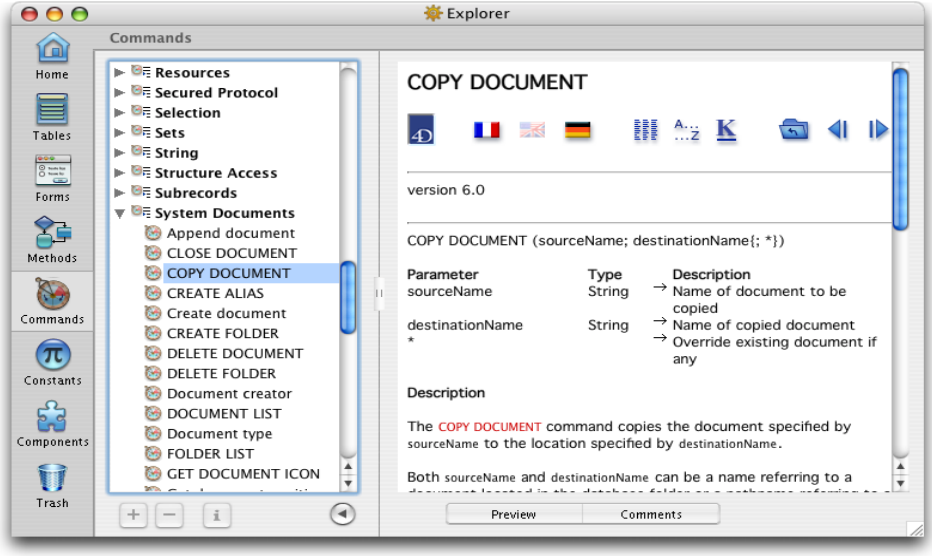
The corresponding HTML page will be displayed in your browser.



HTML documentation pages can be stored on a CD-Rom, a DVD, on your hard disk, or can come directly from the 4D, Inc. Web site. The location from which the page is loaded is defined by the “Documentation Access from the Explorer” parameter in the application Preferences (see the “Documentation Page” paragraph on page 190).

Displaying Documentation in Preview Area (Mac OS Only)

Under Mac OS, when you select a 4D command in the Explorer, its HTML description is displayed in the preview area:

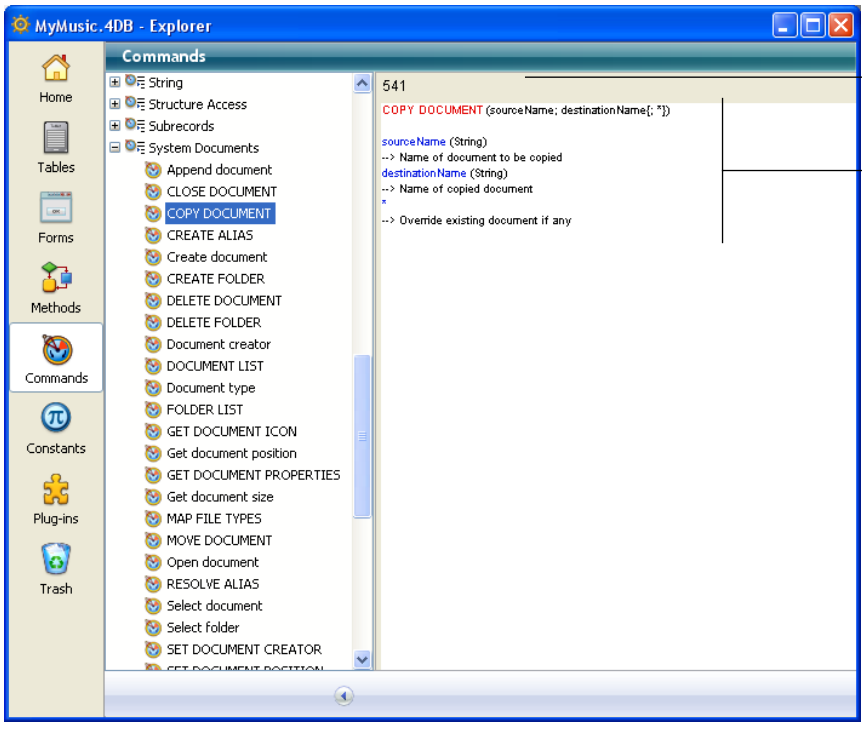


The navigation buttons and hypertext links are active on each displayed page. On the other hand, it is not possible to use the standard Web navigation functions (Next / Back), nor to copy information.

The location from which the page is loaded depends on the parameters set in the application Preferences (see the "Documentation Page" paragraph on page 190).

Displaying the Command Syntax (Windows)

When the preview area is open, it displays the number of the selected command and a description of the command syntax:



Note The command number is used by the Command name function (for more information, refer to the *4D Language Reference* manual).

If you do not want to display the syntax, click on the command name while holding down the **Alt** key.

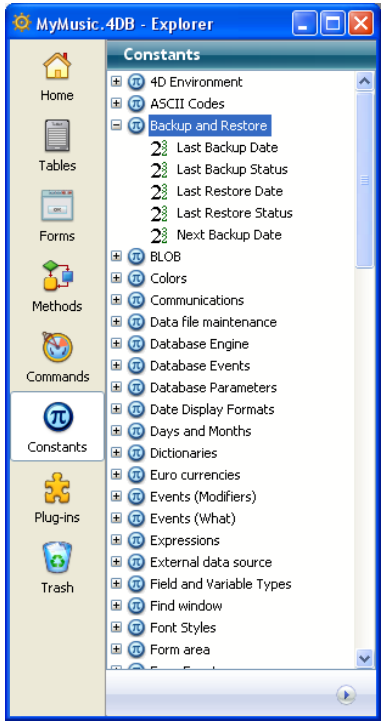
Using Drag and Drop

You can add a command to a method using drag and drop. Highlight the desired command in the Explorer and drag it to the Method editor window. By default, the command and its syntax are inserted at the same time.

If you do not want to insert the syntax, press the **Alt** key (Windows) or **Option** key (Mac OS) when dragging the command.

Constants Page

The Constants page contains a hierarchical list of all the constants that can be used in methods.



If the preview area is displayed, it shows the value of the highlighted constant.

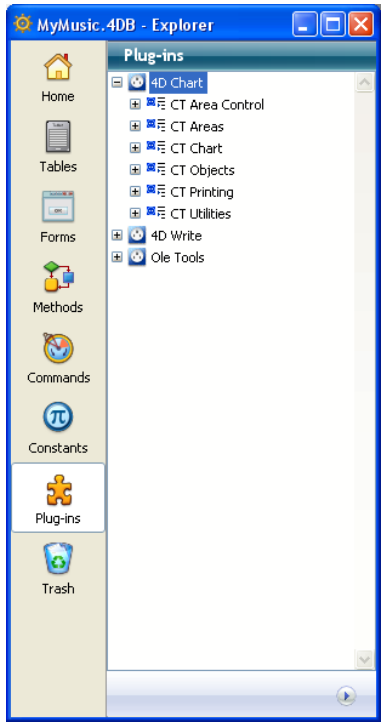
The constants displayed may come from 4D, plug-ins or custom resources. For information on using constants in methods, see the "Constants" section in the *4D Language Reference* manual.

Using Drag and Drop

You use constants frequently in your methods. Instead of typing the constant you can add it to a method from the Explorer. Highlight the desired constant and drag it to the Method editor window. By default, the constants will be underlined when the Method editor parses the line of code.

Plug-ins Page

The Plug-ins page lists all the plug-ins installed in a database as well as their commands, generally grouped by themes.



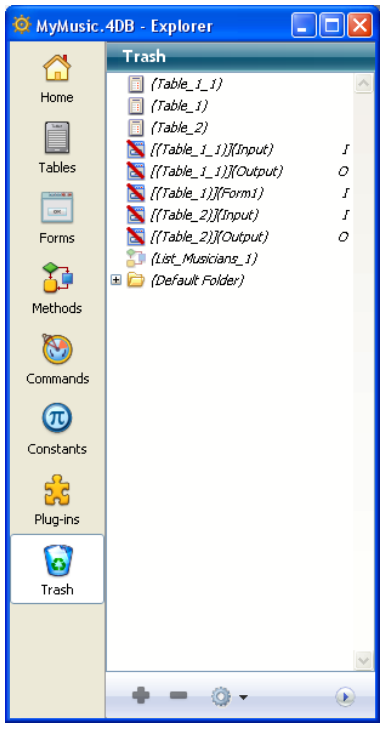
You can insert plug-in commands into your methods using drag and drop from the Explorer.

The preview area displays additional information about the plug-ins as well as the location of the active files.

Note For more information about installing plug-ins in your application, please refer to the "Installing Plug-ins" paragraph on page 47.

Trash Page

The **Trash** page in the Explorer gathers all Folder, Table, Form or Project method objects that have been deleted from the database:



Just like the Windows or Mac OS trash bin, the 4D Trash is an element of interface security that reduces the risk of accidentally deleting items.

Objects placed in the Trash no longer appear in 4D editors or menus and can no longer be modified, used, moved, etc. Other similar objects can be created with the same name.

Objects placed in the Trash can nevertheless still be restored as long as the Trash has not been emptied (see below).

It is also possible to display the objects in the preview area of the Trash page. Moreover, any comments associated with the deleted objects remain visible.

4D Server The contents of the Trash are identical for all client machines.

Deleting Objects

Objects can be deleted from the **Home** page (folders) or the **Tables, Forms** and **Methods** pages.

To delete an object from the Explorer, simply select it and choose the **Delete** command in the Explorer contextual menu (**right-click**), or drag it to the Trash, or click on the delete icon located at the bottom of the page.

On the **Home** page, the Trash can also be selected as a destination in the Move To contextual menu.

A table can also be placed in the Trash. The table then disappears from the database editors, including the Structure editor. When you place a table in the Trash, the forms of the table are automatically deleted (put into the Trash) and their associated methods are erased.

You cannot delete forms with the following attributes: I (current input form), O (current output form) or B (form with both attributes) except when the table of this form is deleted. Similarly, folders containing one of these non-deletable objects cannot be deleted either.

Restoring an Object

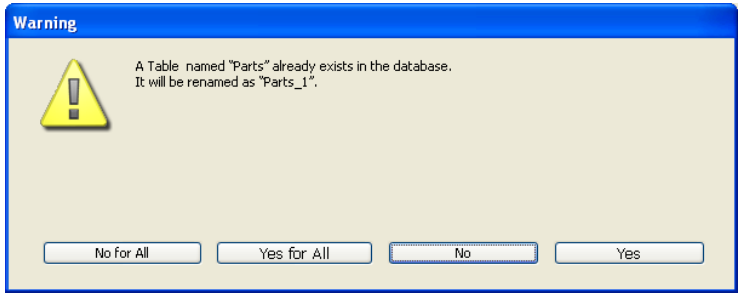
You can restore any object(s) placed in the Trash at any time (as long as it has not been emptied). To do this, simply select the object(s) then drag them to the appropriate page icon at the left of the window or choose the **Move To** > command in the contextual menu of the list:

The **Move To** submenu lists the folders present in the database (for more information, please refer to the "Home Page" paragraph on page 103). If the database does not contain any folders, only the **Top Level** item is available.

When you choose a command from this submenu, the object selection is restored and replaced in the chosen folder. If you choose **Top Level**, the selection is placed at the first level, not in any folder.

If one or more objects of the database have the same name as one or more object(s) that you wish to restore, 4D displays an alert dialog box offering to:

- Replace the object in the database (forms, methods, etc.).
- Rename the object to be restored *name_1* (tables).



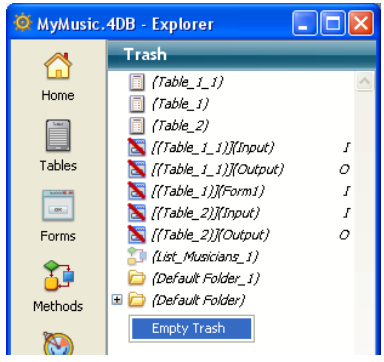
The **Yes** button renames the object being restored or replaces the existing object. In the same way, the **Yes to all** button renames or replaces all the objects of the selection (when applicable). If you click **No** or **No to all**, the restoring of objects is not carried out.

Note Only the designated object is restored. More specifically, forms deleted with a table are not automatically restored with the table.

Emptying the Trash

By default, deleted objects remain in the Trash until it is emptied. You can decide to empty the Trash at any time, for example, before compacting, in the interest of reducing the size of the structure file. All objects in the Trash are then permanently deleted from the database.

To empty the Trash, choose the **Empty Trash** command in the Explorer contextual menu (**right-click**) or in the options menu of the page:



An alert dialog box indicating the number of “deletable” objects present in the Trash allows you to confirm or cancel the operation:

Click **OK** to delete all the objects in the Trash.

Using Comments

The Explorer allows you to write comments about objects in your database. Using comments is particularly appropriate for databases being developed by multiple programmers and is generally good programming practice. Your comments are displayed in the preview area of the Explorer.

The following objects accept comments:

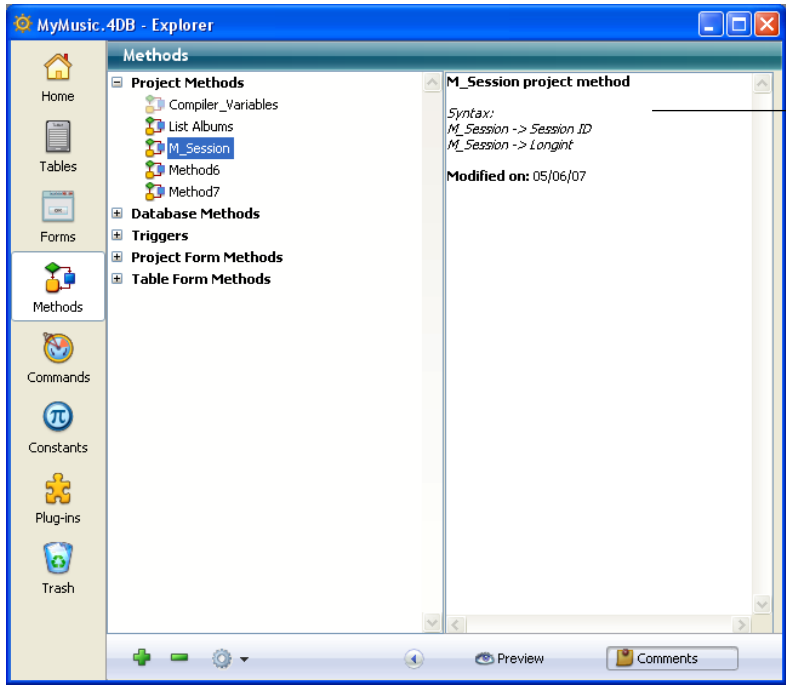
- Folders
- Methods (database methods, component methods, project methods, form methods and triggers),
- Forms,
- Tables,
- Fields,
- Plug-ins.

A comment can be entered as styled text (i.e., the characters can have different font styles or colors, etc.) that can be modified and viewed at any time in the Design environment. It can contain a description of an object with which it is associated as well as any information necessary to understand how the object functions in the database. The comments you create are stored in the database’s structure.

Moreover, 4D allows you to generate automatic comments, which means that 4D automatically enters comments when an object is created or modified.

Associating a Comment to an Object

You create, view, and modify comments from the Explorer. To access the Comments area for an object, select the object and then click the Comments button located below the preview area.
When the **Comments** option is selected, the preview area is replaced by the Comments area.



You can enter up to 32 KB (32,700 characters) of text for each object. The Comments area has a vertical scroll bar that allows you to scroll through the text.

4D Server The Locking icon , located in the bottom left corner of the area, indicates whether the comment is already being edited by another user. If this is the case, the pencil has a slash through it and the comment can only be viewed.

The text is saved as soon as you click outside of the entry area. You can use the standard text editor commands (**Copy**, **Paste**, **Select All**, etc.) available in the contextual menu or by using keyboard shortcuts in the Comments area. You can also navigate the text in the comments area by using navigation keys as you would for any other text area.

Modifying the Style of the Comments

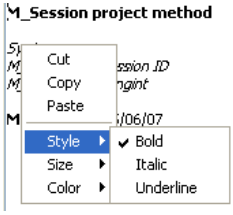
You can enhance the style of the comments (add bold or italics) and change the font attributes (font, font style, font size, or color) using a contextual menu.

► To modify the style of the comments' text:

1 **In the Comments area, select the text that you want to modify.**

2 **Click in the area with the right mouse button.**

A hierarchical pop-up menu appears:



3 **Select the attributes that you want to apply to the text.**

Inserting Automatic Comments

It is possible to activate an automatic commenting system that can be used for methods and forms in the database. When this system is activated, a comment is automatically associated with every method or form created or modified in the database. An automatic comment can consist of both static text (such as "Modified by") and variables (such as the current date, current time, and current user as specified in the 4D passwords).

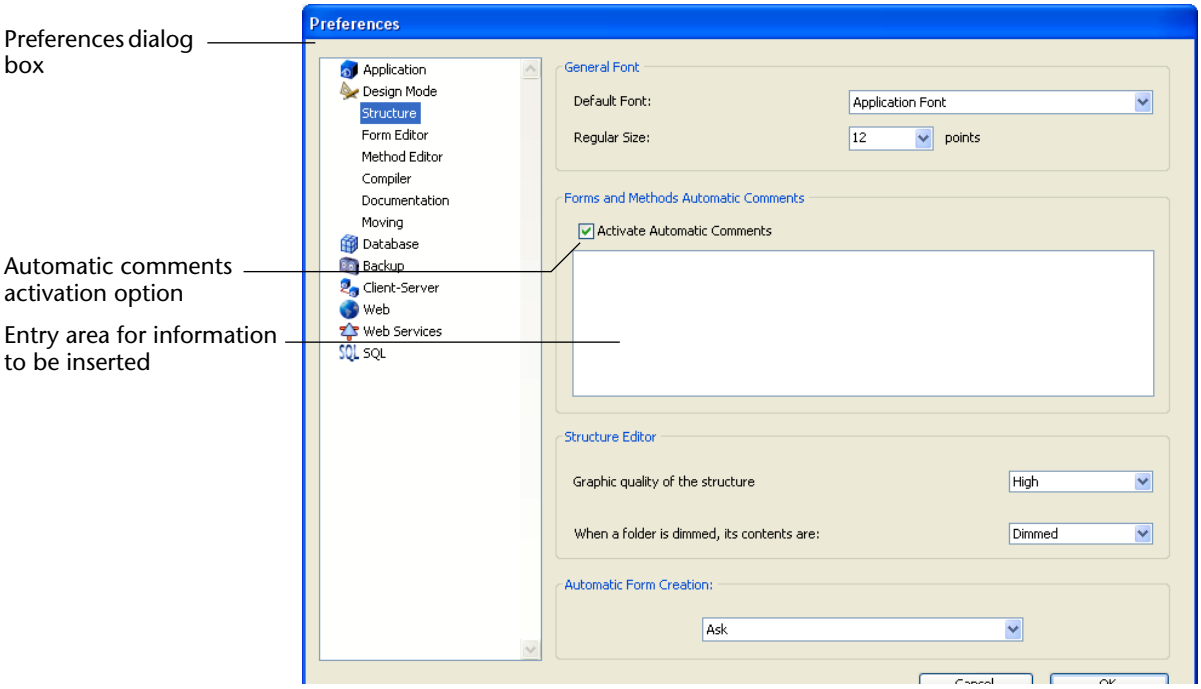
Automatic comments are defined in the Preferences dialog box of the application. As with standard comments, you can view them using the Explorer.

► To activate the automatic commenting system:

1 **Select the Preferences... command in the Edit menu (Windows) or the application menu (Mac OS).**

2 **In the Preferences dialog box, display the Structure page of the "Design Mode" theme.**

3 Select the "Activate Automatic Comments" option:



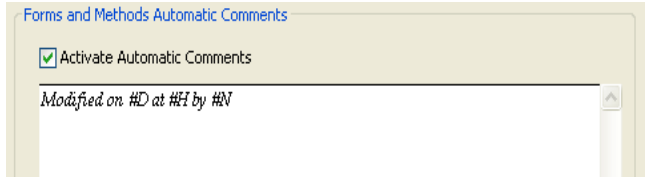
4 Enter the set text of the comments.

You can insert static text and variable elements.
To insert variable elements, you can use the contextual hierarchical pop-up menu. You can also directly enter the following variables:

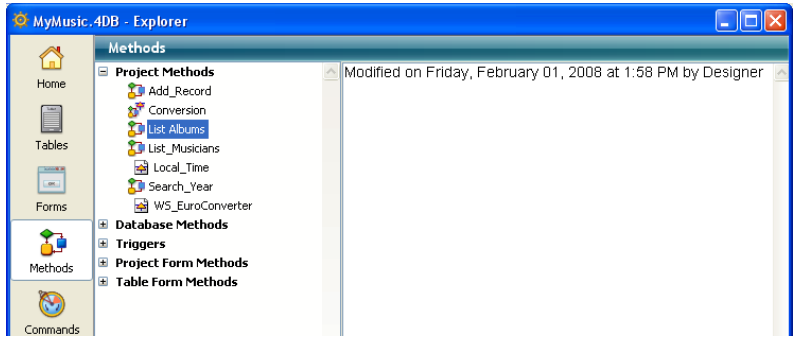
- **#D** for the date
- **#H** for the time
- **#N** for the current user.

Note If your database doesn't have a password system, #N returns the string "Designer."

For example, if you enter the following values:



The automatic comment will be added to the comments for all new methods and forms, as well as all existing methods and forms that are modified after automatic commenting is turned on:



4D Server Automatic comments can be modified by any client workstation that has access to the Preferences. You can also modify these parameters in the Preferences of the server machine.
Every modification made to the Automatic Comments page is immediately taken into consideration by each client workstation as soon as an object is modified and its comments are accepted.

Runtime Explorer

The Runtime Explorer window allows you to view the behavior of the different structural elements in your database and to verify that the available resources are operating as expected. The Runtime Explorer is particularly useful in your database's development and analysis phase.

Displaying the Window

The Runtime Explorer window can be displayed in the Design and Application environments, in both compiled or interpreted mode.

Note Only the Designer and Administrator have access to the Runtime Explorer window.

The Runtime Explorer can be displayed in two types of windows: in a standard window or in a floating palette. The floating palette always remains in front of other open windows.

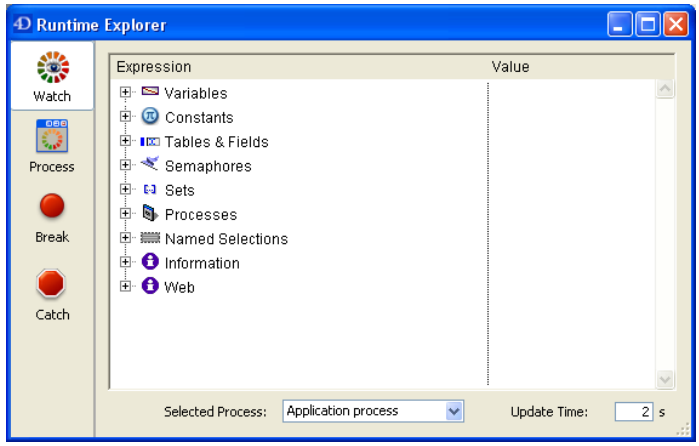
► To display the Runtime Explorer in a standard window:

1 Choose Runtime Explorer from the Run menu.

► To display the Runtime Explorer as a floating palette:

1 **Under Windows, press Ctrl+Shift+F9.**
Under Mac OS, press Command+Shift+F9.
OR
Hold down the Shift key and choose Runtime Explorer from the Run menu.

The Runtime Explorer window has four pages that are accessed via buttons on the left-hand side: **Watch**, **Process**, **Break,** and **Catch**.



Watch Page

The **Watch** page is a debugger and displays information about code execution concerning the application and the selected process

- **Selected Process**: This drop-down list contains all the processes that are being executed in the database. It allows you to select the process(es) that you want to observe.
- **Update Time**: In this area, you can set a value (in seconds) that indicates how often the information on the page will be updated.

The “Expression” column displays the names of the objects and expressions. The “Value” column displays the current value of the objects and expressions. These columns can be resized, one in relation to the other. To do so, click on the separation line and drag it to the left or right.

By clicking on a value in the right column, you can modify the object’s value, if the object allows this.

The multi-level hierarchical list is organized by theme. The themes are as follows:

- **Variables**: Allows you to view the list of the database's interprocess variables as well as the list of the selected process's process variables.
- **Constants**: Allows you to view the list of constants defined in the database.
- **Tables and Fields**, **Semaphores**, **Sets**, **Processes** and **Named Selections**: The information provided in these themes is identical to the information provided by 4D's debugger. For more information, refer to the "Debugger" chapter in the *4D Language Reference* manual.
- **Information**: This theme displays general information concerning database operation, such as the current default table (if one exists), available memory, query destination, etc.
- **Web**: Displays information concerning the Web server of the application (only available if the Web server is active): **Web File To Send**, **Web Cache Usage** (number of pages present in Web cache as well as its use percentage), **Web Server Elapsed Time** (duration of Web server use in hours:minutes:seconds format), **Web Hits Count** (total number of HTTP requests received since Web server launch, as well as the instantaneous number of requests per second) and **Number of active Web processes** (all Web processes together).
  The expressions contained within this theme cannot be modified.

  To delete an expression or a theme, select the corresponding line and press the **Delete** or **Backspace** key.

  You can also add a **New Expression** or a 4D **Command**, or perform global actions: **Delete All**, display all the **Standard Expressions**, **Collapse All** or **Expand All**.

To do this, select the corresponding command in the contextual menu which appears when you click the **right button** of the mouse in the window:

Note You can add a new expression by double-clicking in the Expression column.

In addition, several display options are available in the lower part of the contextual menu.

- **Show Types**: Displays or hides the types of fields next to their names in the list of tables & fields.
- **Show Field and Table Numbers**: Displays or hides the table and field numbers next to their names in the list of tables & fields. For each field, the following format is applied: [TableNum]FieldNum.
- **Show Icons**: Displays or hides the object icons in the hierarchical list.
- **Sorted Tables and Fields**: Sorts the list of tables and fields by alphabetical order (by default, these objects appear in the order that they were created).
- **Show Integers in Hexadecimal**: Displays the variables declared as Integer or Long Integer types in their hexadecimal form.
- **Enable activity monitoring**: Displays additional information concerning the scheduler and the communications network. This low-level information, grouped in the **Scheduler** and **Network** items, allows advanced monitoring of the internal activity of the application. Be careful, activating this option slows down processing.

Process Page

The **Process** page allows you to graphically view the CPU time consumed by each process as well as the state of each process.

You can show or hide the graphical display of each process by clicking on the expansion icon to the left of the process's name.

The use of this page is described in the chapter "Processes" on page 929.

Break and Catch Pages

The **Break** page allows you to view and manage break points that you have placed in your code.

The **Catch** page displays the break points set in the database in relation to commands (or expressions).

The general functioning of these pages is identical to the debugger's **Break List**. For more information, refer to the "Debugger" chapter in the *4D Language Reference* manual. The commands in the **Break List** menu are replaced by the buttons in the window's toolbar.

Note You can also set break points directly in the Method editor (refer to "Method Editor Window" on page 781).

Searching and Replacing in the Design

4D provides a search and replace function for objects in all of the design environment. You can search for a string or a type of string in part of or in the entire database structure on the basis of custom criteria ("starts with", "contains", etc.). The results are displayed in a window, where it is possible to prefix or rename the objects found.

You can, for example, search for the string "MyVar" only in methods, then automatically add the prefix "Perso_". The search can go as far as searching all the form objects in the database.

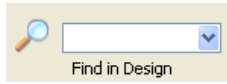
You can carry out a "quick" search directly from the toolbar of 4D, or an "advanced" search, using the search window.
Replacing is always carried out using the results window.

Quick Search

The quick search is based on searching for a character string.
It is carried out using the toolbar of the 4D Design environment. If the toolbar is hidden, you must use the Find in design window.

▶ To carry out a quick search:

1 Enter the character string you want to search for in the toolbar of the Design environment:



The quick search is done without options:

- Search for all types of objects,
- Search of the “is exactly” type which is not case sensitive,
- Search among all the database objects.

OR
If you want to repeat a search that was already done during the session, you can select it in the scroll-down list.

If the search was performed using the Find window, it is carried out using any options set in this window.

2 Press Return or Enter.

The search begins immediately. When it is done, a result window appears, listing all the objects containing the string entered (see the “Results Window” paragraph on page 146).

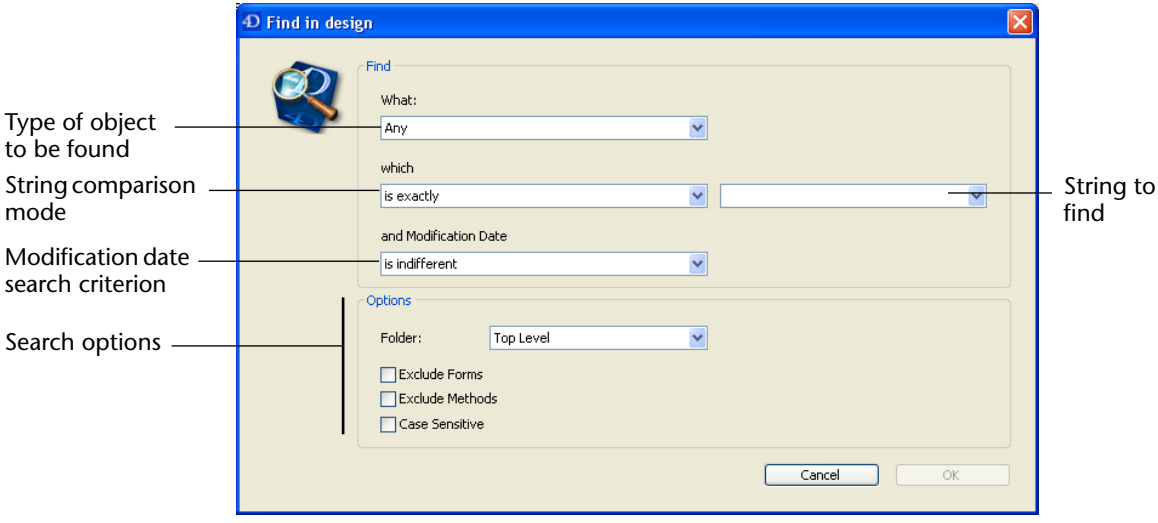
Find in Design Window

If you want to use advanced search criteria, you must use the Find in design window. Keep in mind that it is also possible to launch simple searches using this window.

► To carry out a search using the Find in design window:

1 Click on the magnifying glass button in the toolbar of 4D
OR
Choose the Find in Design... command from the Edit menu.

The Find in design window appears:



The areas of this window vary dynamically depending on the selections made in the menus.

2 In the designated area, enter a string to find.

Note The at-sign (@) can be used as a "wildcard," for example to specify searches of the "starts with" and ends with" type (e.g.: "m@ base").

3 If desired, specify the object type to which you want to restrict the search by selecting it from the What menu.

- Restricting the search to a certain object type makes the search faster.
- If you don't want the search to be limited to an object type, choose **Any** in the **What** menu. In this case, the search will only be carried out on the basis of the object name, its modification date or a combination of both.
- If you select **Expression** in the **What** menu, it is not possible to specify a comparison mode and the **Edit Expression** button appears. It allows you to specify the expression to search for directly in the Formula editor. For more information about this editor, refer to the "Formula Editor" paragraph on page 981.

The types of character strings that you can search for are described in the "String Types and Scope of Search" paragraph on page 144.

4 Specify (if necessary) the string comparison mode.

This menu provides the "contains," "is exactly," "starts with" or "ends with" options.
If you do not want to use the object name as search criterion, choose the "is indifferent" option. In this case, the search is carried out only on the basis of the type of object, or its modification date, or a combination of both.

5 Specify (if necessary) the modification date criterion.

This criterion is valid for searches among methods and forms only. It can be used alone or combined with one or more other criteria.
If you do not want to use the date as a search criteria, choose the "is indifferent" option. In this case, the search will be carried out only on the basis of the type of object, or its name, or a combination of both.

6 Set the search options (if necessary).

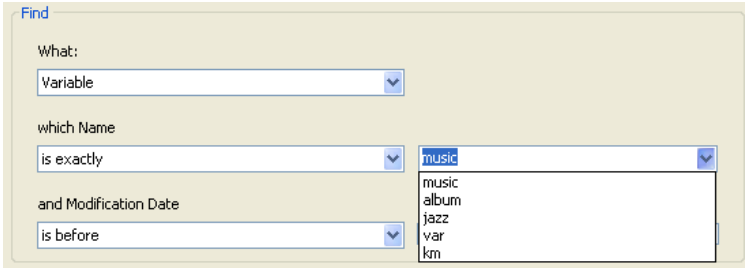
These options are described in the "Searching Options" paragraph on page 145.

7 Click OK or press the Enter key.

The search begins. When it is finished, the result window appears, listing the objects containing the string entered (see the "Results Window" paragraph on page 146).

Note You can interrupt a search that is underway at any time by hitting the **Esc** key.

Once you have executed a search, the value entered in the search area is saved in memory. This value, as well as all the other values entered during the same session, can be selected from the combo box:



Using the combo box, you can quickly do the same search many times.

String Types and Scope of Search

The Find in design dialog box allows you to limit the search to a particular object type or to search across all object types:

- An expression, for example "total:=Sum([Accounts]Total)"
- A variable, for example "$vpPicture1"
- A table or field name, for example "[Clients]Name"
- A form object name, for example "Background"
- A comment, for example "Modified on."
- A project method name, for example "M_Add"
- If you leave the "What:" area blank, you can also search for a type of string.

Note When the "Project Method" type is selected, the "Exclude Methods" option disappears. Moreover, in this case the results window then includes the "Callees" and "Callers" lists (see the "Callees and Callers Page" paragraph on page 148).

By default, the search will be done throughout the entire structure of the database. You can, however, exclude methods and/or forms from the search. Depending on the designated object type, the search will be carried out among the following types of objects:

- Forms (which can also be excluded)
- Methods of any type (which can also be excluded)
- Menus and menu commands
- Lists
- Tables and fields
- Comments
- Pictures from the Picture library
- Static texts
- Help tips
- Formats/Filters
- Style sheets
- Plug-in commands
- 4D commands
- Constants
- Folders.

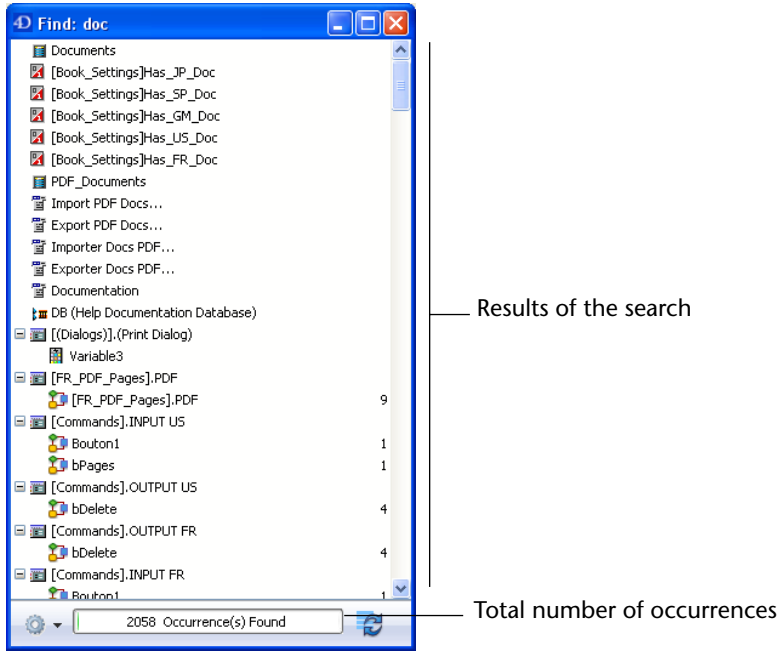
Searching Options

You can select various options that can help speed up your searches. Certain options are unavailable depending on the type of search requested:

- **Folder**: This option restricts the search to a certain folder. By default ("Top Level" option), the search takes place in all the folders.
- **Exclude Forms**
  When this option is selected, the search is done throughout the database, except in forms and form names.
- **Exclude Methods**
  When this option is selected, the search is done throughout the database, except in methods and method names. This option is hidden when the "Project Method" type of object is searched for.
- **Case Sensitive**
  When this option is selected, the search takes the case of the characters as they have been entered in the Find dialog box into account. Therefore, if you search for "MyVar," 4D won't find "myVar."

Results Window

Once the search has finished, the occurrences found appear in a new window and are shown as a hierarchical list organized by type of object found. It also includes an options menu that can be used to carry out various operations on the objects found.



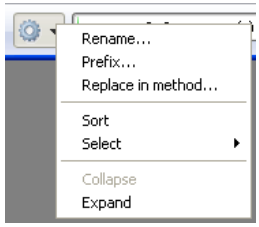
You can double-click on a line in this window to view the object in its editor. If you do several searches, each search opens its own result window, leaving previous result windows open.

The window includes a contextual menu that can be used to carry out various standard operations:

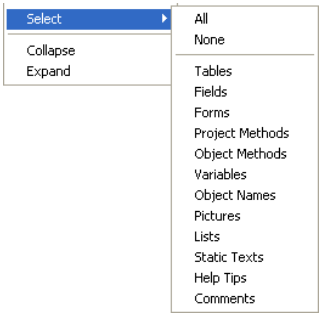
When more than one occurrence of an object or string has been found within a method, the list indicates their number next to the object name:

Options Menu

The commands of the options menu are enabled or disabled depending on the objects selected. For example, if the selection contains 4D commands, the renaming and prefixing functions are disabled:



- **Rename**: This command can be used to rename the selected object. This operation is detailed in the following paragraph.
- **Prefix**: This command can be used to prefix the names of all the objects selected in the list. This operation is detailed in the following paragraph.
- **Replace in method**: This command can be used to carry out a search and replace among one or more methods. This operation is detailed in the following paragraph.
- **Sort**: This command sorts the contents of the window by alphabetical order. Holding down the **Shift** key while selecting this command will sort in reverse alphabetical order.



Select: This submenu is used to make automatic selections among the objects shown in the window (you can also manually select one or more objects using the standard **Shift+click** or **Ctrl/Command+click** shortcuts). It is necessary to make a selection among the objects of the list in order to be able to rename or prefix this selection.
Choose **All** to select all the contents of the window. You can also use the standard **Ctrl+A** (Windows) or **Command+A** (Mac OS) shortcut, or use the contextual menu of the list.

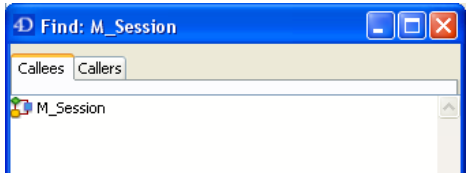
- **Expand/Collapse**: These commands can be used to respectively expand or collapse all the hierarchical items in the list of results.

Redo Search

This button  lets you redo the search with the same criteria and options. This can be helpful, for example, to make sure that all the desired replacements have been carried out.

Callees and Callers Page

When the search has been carried out only among project methods, the results window will include two lists that can be accessed using their respective tabs: **Callees** and **Callers**:



- The **Callees** list displays the project methods found by the search, in other words, those whose name and/or modification date correspond to the search criteria.
- The **Callers** list displays the objects (methods, forms and so on) that call the project methods found by the search.

Note Objects calling a project method can be searched for directly from the Explorer or the Method editor. For more information, please refer to the "Searching for Callers" paragraph on page 122.

Prefixing and Renaming

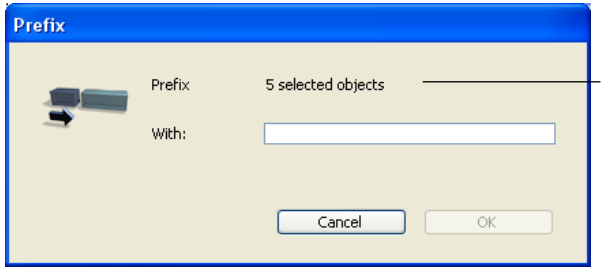
The search results window can be used to prefix and rename objects, or to replace a string in a selection of methods.

Prefixing

Prefixing can be helpful when you want to enhance the readability of your code or provide a quick means for locating the object families of your database.

The prefixing command of the options menu is enabled when the selection contains only objects that can be prefixed. Only **Variables**, **project methods** and **forms** (table and project) can be prefixed.

When you select this command, the following dialog box appears:



The entry area is where you enter the character string to be used to prefix the selected objects.

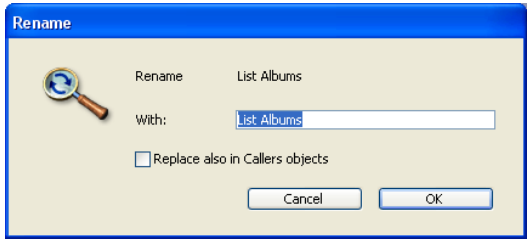
- **Replace also in Callers objects** (option appears when you prefix objects in the Callees list): Check this option if you want for the prefix to also be applied to the objects that call the selected objects, for example the methods calling the method being prefixed. This option is used to maintain proper code operation after the prefix is applied.

Note Since object names are limited to 31 characters, it may happen that adding a prefix makes the object name too long. In this case, a warning message will appear that allows you to cancel the operation.

Renaming

This operation consists in renaming a selected object. The **Rename...** command of the options menu is only enabled when the selection contains a **single object** and when this object can be renamed. **Variables**, **project methods** and **forms** (table and project) can all be renamed.

When you click this button, the following dialog box appears:



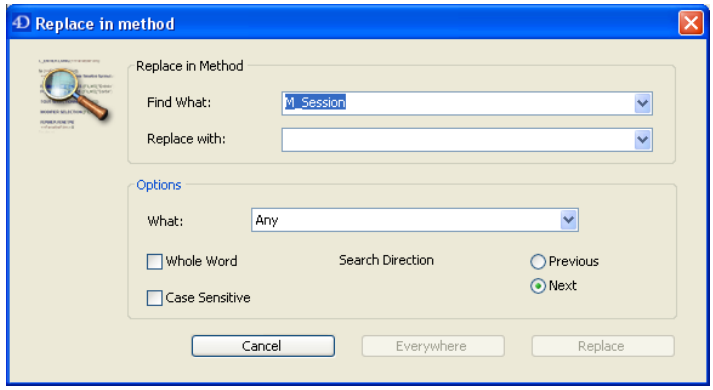
The entry area is where you enter the new name of the selected object.

- **Replace also in Callers objects** (option appears when you prefix objects in the Callees list): Check this option if you also want the object to be renamed in the objects that call it. This option is used to maintain proper code operation after the object is renamed.

Replacing in Methods

This operation consists in carrying out an overall “search and replace” among one or more methods. The **Replace in method...** command of the options menu is only enabled when the selection contains one or more methods.

When you click this button, the standard Find and Replace dialog box for methods appears:



This dialog box can be used to find and replace a character string within methods. For more information about its operation, please refer to the “Find and Replace in Methods” paragraph on page 821.

Drag and Drop of Objects in Design Environment

In 4D, you have the possibility of moving objects between two applications in the Design environment. You can thus recopy the tables, forms, methods, etc. that you have created in one database into another. This can greatly facilitate and accelerate database development.

Moving objects is not limited to individual objects, it can also concern any objects that are referenced by the object, i.e. its dependent objects. You can thus move entire functionalities. For example, if you have created a custom query dialog box, you can move the form used for the dialog box as well as all the methods, pictures and other objects it uses. You can copy the form into another database or into a library grouping the functionalities currently used in your databases.

Certain objects are also inseparable: they are mandatorily moved along with their “parent” objects. The list of inseparable objects is provided below.

Movable Objects

Objects can be moved from the Tool box, the Explorer and the Form editor. Moreover, it is possible to carry out a move from the results window of a Find in design search (see the "Results Window" paragraph on page 146).

For structural consistency, copying certain objects will lead to the copy of any objects that are inseparable from them. For example, copying a form will lead to the copy of any form method and/or object methods that are attached to it. These inseparable objects cannot be moved directly on their own.

Here is a list of movable objects along with their inseparable objects:

Movable object	Inseparable objects
Tool box	
Lists	
Style Sheets	
Formats/Filters	
Pictures from Library	
Help tips	
Explorer	
Project forms	Form methods
Table forms	Form methods
Project methods	
Folders / Subfolders	
Tables	Fields, triggers
Form editor	
All form objects (buttons, variables, etc.). When moving a form, all the objects it contains are moved with it.	Object methods

How Moving Works

Objects can be moved either using standard drag and drop or by copy/paste.

To move objects between two databases using drag and drop, you must duplicate your version of 4D or use 4D and a 4D Client.

In the case of inter-database moving, moved objects can be pasted or dropped into the same environment as that of their departure (Tool box, Explorer, etc.) or into other areas of the application. 4D will carry out the appropriate action according to the context, whenever possible. For example, it is possible to drop a form onto the Method editor window; in this case, the name of the form is inserted into the method.

During a move, if an object of the same type and with the same name already exists in the destination database, by default the existing object will be replaced by the moved object. The moving dialog box will be displayed in this case; it indicates the objects that will be replaced and lets you modify this action.

The following mechanisms should be noted:

- **Views and level**: Form objects that are moved keep the same location properties as in the editor, in particular their position in the views or in the different levels of the form.
- **Inherited forms**: Inherited forms are not moved with the source forms, however their reference is kept. Moreover, inherited forms are considered as dependent objects and it is possible to use another (existing) form as inherited form when the move is carried out (see the "Dependent Objects of Forms" paragraph on page 152).
- **Access rights**: Forms and project methods that are moved do not keep their original access rights. They are automatically given the default value ("All groups").
- **Folders**: When you move a folder from the Home page of the Explorer, the operation includes the folder and all of its contents (tables, forms and project methods), which can represent a considerable volume of data. During this type of move, a warning dialog box will appear to indicate this functioning.

Note It is not possible to drag objects from the **Trash** page of the Explorer.

Dependent Objects of Forms

A form (table or project) can refer to various other objects like lists, pictures, etc.: these objects are called *dependent objects*.

In certain cases, you may need to move all the dependent objects whereas in others, you may only want to move some or even none of these objects.

4D lets you control the moving of dependent form objects via moving Preferences as well as a specific Moving dialog box.

Moving Preferences specify the principles to be applied for moving dependent objects. You can choose various default options for each type of object. For more information, please refer to the "Moving Preferences" paragraph on page 157.

Moving Dialog Box

When you move a selection of objects (by drag-and-drop or copy-paste) between two 4D databases or between a database and an object library, you have the possibility of displaying a dialog box that lists all the objects being moved as well as the actions that will be associated with them in the destination database.

This dialog box is entitled "Moving dialog box" and it is displayed when at least one of the following cases is true:

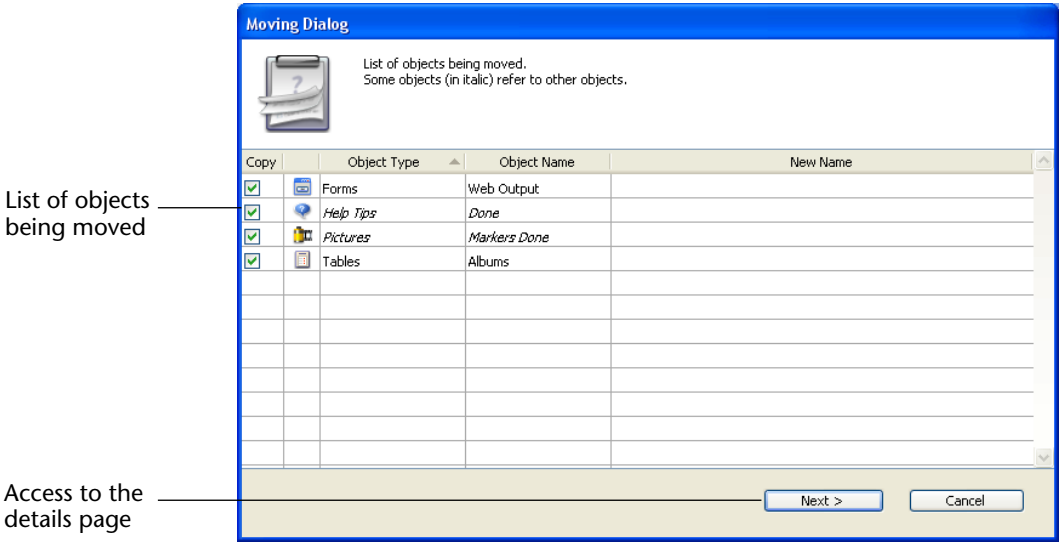
- The "Always display" option is selected in the moving preferences (see the "Moving Preferences" paragraph on page 157).
- At least one of the objects being moved has a name conflict with an object in the destination database.
- The **Use another object** default action has been selected for at least one type of dependent object that is being moved.

Apart from these cases, if the moving of the objects does not cause any conflict, the moving dialog box will not appear and the objects will be copied directly.

This dialog box allows you to view and/or modify the moving settings depending on the context. It includes two pages: the **main page** and the **details page**. You can switch between these pages using the **Next>** and **<Back** buttons.

Main Page

The main page displays the list of objects being moved:



The objects that have a name conflict appear in **bold** and the dependent objects appear in *italics*. The different columns indicate the type and name of each object as well as, if necessary, its "new" name, in other words, its default name in the destination database. You can modify this name on the details page if desired (see following paragraph).

The checkbox in the "Copy" column indicates whether the object will or will not be copied into the destination database. This option is active: you can uncheck it in order to quickly resolve any copying conflicts involving a specific object. Note that if a conflict involves a dependent object, its reference (name) will be kept in the destination database. You can alternately check/uncheck all the boxes in this list using **Alt**+click (Windows) or **Option**+click (Mac OS) on any box. You can also use the contextual menu of the dialog box.

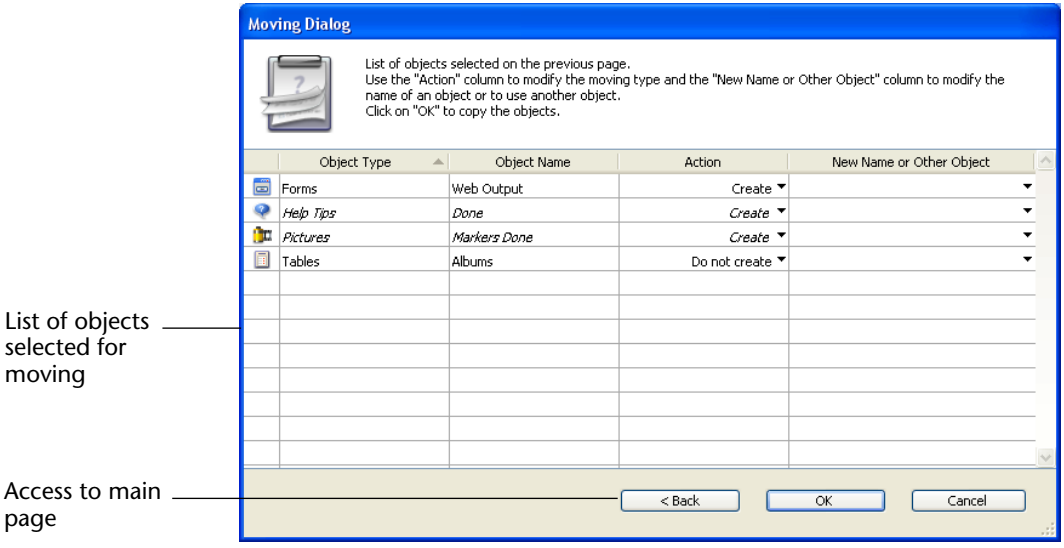
If the default moving settings suit you, you can click directly on **OK** in order to proceed with the moving of the objects.

Note If at least one object is associated with the **Use another object** action, you cannot validate the copy without having specified the target object in the destination database (or deselected this object).

If you want to modify certain actions, click on the **Next>** button so as to display the details page. If you want to cancel the moving of the objects, click on **Cancel**.

Details Page

The details page lists the objects to be copied (those that were left checked on the main page) and can be used to modify the moving settings:



You can use the pop-up menu of the “Action” column to modify the actions carried out on the objects. The actions provided in the menu depend on the type of object selected and are described below.
It is possible to modify the action assigned to several rows in a single operation. To do so, simply select the rows to be modified, then choose a new action in the “Action” column of one of the selected rows. The modification is then carried out in all the rows of the selection where it is applicable. If the action is incompatible with one of the rows, that row is not modified and a warning dialog box informs you of this.

The “New Name or Other Object” column displays the name that will be attributed to the object once it is copied into the destination database. You can change this name (be careful not to use a name that already exists in the destination database since this would create a new name conflict).

In the case of dependent objects, this column also lets you designate another object of the destination database (when the **Use another object** action is selected). For example, in the case of moving a table form, you can designate a table of the destination database as the table to which it will belong instead of creating the table.

Possible Actions

The following alternative actions are possible:

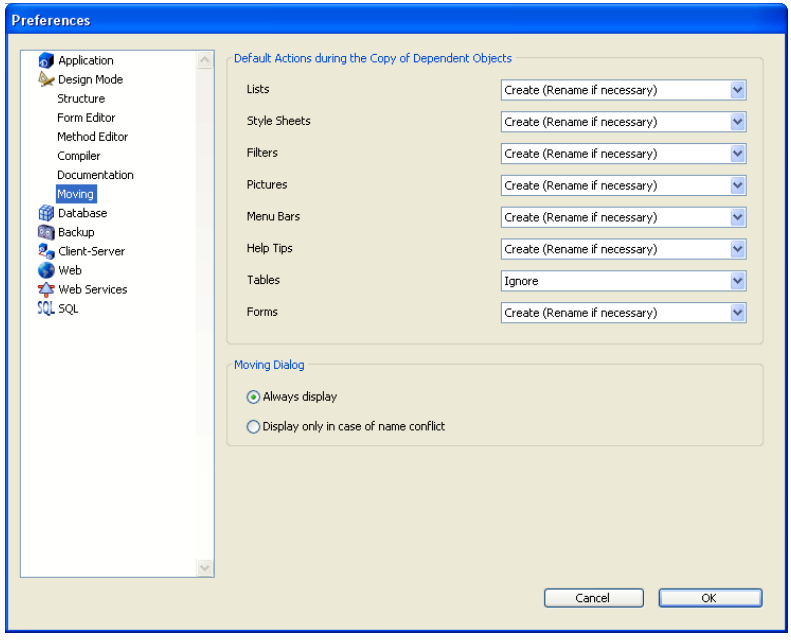
- **Do not create**: The object is not copied. In the case of a dependent object, its reference (name) is kept if an object with the same name already exists in the database (in this case, it is used by the main object). If no object with the same name is available, the reference will be deleted.
- **Replace**: This option is proposed when an object of the same type and with the same name already exists in the database. In this case, the object in the destination database is replaced by the one of the departure database.
- **Create**: The dependent object is copied into the destination database with its properties (option proposed when there is no name conflict).
- **Create and rename**: This option is proposed when an object of the same type and with the same name already exists in the destination database. By default, the object is renamed by adding a number as a suffix. In this case, you can rename the moved object in the "New Name or Other Object" column. Naturally, the object references are updated in the destination database.
- **Use another object**: This option is only available for dependent objects. It lets you use another object already present in the destination database as reference. In this case, the "New Name or Other Object" column contains the list of other objects that can be used.
- **Use table with the same name**: This option is proposed when a table with the same name already exists in the database. In this case, the "New Name or Other Object" column contains the list of tables that can be used instead of the dependent table.

If dependent objects themselves reference other objects, the list is updated according to your settings.

If the moving settings suit you, you can click on **OK** to proceed with the moving of the objects. Click on **<Back** to return to the main page. If you want to cancel the moving of the objects, click on **Cancel**.

Moving Preferences

You can pre-configure the moving of objects in the database via the Design/Moving page of the Preferences:



These preferences will be applied to the dropping/pasting of objects into the database, in other words, when the database is used as a destination database.

Default Actions During the Copy of Dependent Objects

The upper part of the dialog box can be used to configure the moving of dependent objects, in other words, the objects linked to the forms being moved (see the "Dependent Objects of Forms" paragraph on page 152). You can set an action for each type of dependent object.

These default actions will automatically be applied if the moving of the objects does not cause any conflicts and if the **Display only in case of name conflict** option is selected (see following paragraph). Otherwise, they will be selected by default in the moving dialog box.

The **Ignore**, **Create (Rename if necessary)**, **Create (Replace if necessary)** as well as the **Use another object** actions are provided for each type of object. Other more specific contextual actions are provided in the moving dialog box when it is displayed. For more information about these actions, please refer to the "Possible Actions" paragraph on page 156.

Below is a description of these options:

- **Ignore**: A dependent object of this type is never copied into the destination database. In the moving dialog box, the **Do not create** action is proposed by default.
- **Create (Rename if necessary)**: A dependent object of this type is always copied into the destination database. In the moving dialog box, the **Create** action is proposed by default if the object does not already exist in the destination database.
  In the case of a name conflict with an object in the destination database, the object being copied is renamed by adding the suffix "_X," in accordance with the principle applied to the main objects (see the "How Moving Works" paragraph on page 151). In this case, the **Rename** action is proposed by default in the moving dialog box.
- **Create (Replace if necessary)**: A dependent object of this type is always copied into the destination database. In the moving dialog box, the **Create** action is proposed by default if the object does not already exist in the destination database.
  In the case of a name conflict with an object in the destination database, the object being copied replaces the existing object. In this case, the **Replace** action is proposed by default in the moving dialog box.
- **Use another object**: This option causes the systematic display of the moving dialog box, even if the "Display only in case of name conflict" option is selected. When moving objects, you must designate an object of the destination database to use instead of the dependent object being copied.

Note These options are only taken into account for dependent objects. For the objects being moved, the default action is of the **Create (Rename if necessary)** type.

Display of Moving Dialog Box

This option can be used to configure the display of the moving dialog box.
When the **Always display** radio button is selected, the dialog box appears each time objects are being moved, which permits more precise control over the operation. If the **Display only in case of name conflict** radio button is selected, the dialog only appears when an object being moved (whether a dependent object or a main one) has a name conflict with an object of the destination database.

4D Server Considerations

4D Server supports the continuing development of a database that is currently in use by allowing you to modify the design of a database in the Design environment while users continue to work with the database in the Application environment. Changes to the design are visible to users as soon as they are saved (when you close the editor window or choose **Save *EditorName*** from the **File** menu).

In addition, 4D Server allows several users to work on the database design concurrently. 4D Server prevents users from erasing each other's changes by providing an object-locking system in which structure objects are locked while they are being modified. Structure objects include the following: field definitions, table definitions, forms, methods, menus, passwords, preferences, and lists. During the time that an object is locked, other users cannot modify it. An object is unlocked when it is closed.

For more information about how object locking works with a particular structure object, refer to the appropriate chapters in this manual.

3 Setting Preferences

This chapter describes each parameter of the Preferences dialog box of 4D.

Access to the Preferences

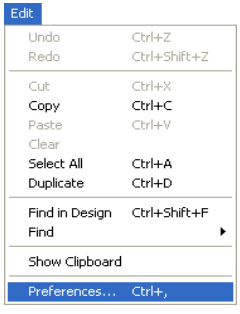
You can set application Preferences in the Design or Application environment. Certain preferences will only apply to the open database while others will apply to all databases opened with your 4D application.

Most of the specifications that you select in the Preferences dialog box take immediate effect except for those that cannot take effect until the database is opened again (such as Startup environment).

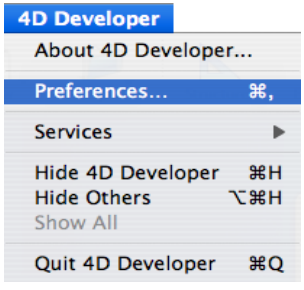
4D Server Object locking occurs when two or more users try to modify the settings in the Preferences dialog box at the same time. Only one user can use the Preferences dialog box at a time.

► To display the Preferences dialog box:

1 **Under Windows: Choose Preferences from the Edit menu.**
Under Mac OS: Choose Preferences from the "application" menu.

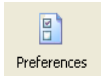


Windows



Mac OS

OR
Click on the "Preferences" button of the 4D toolbar.

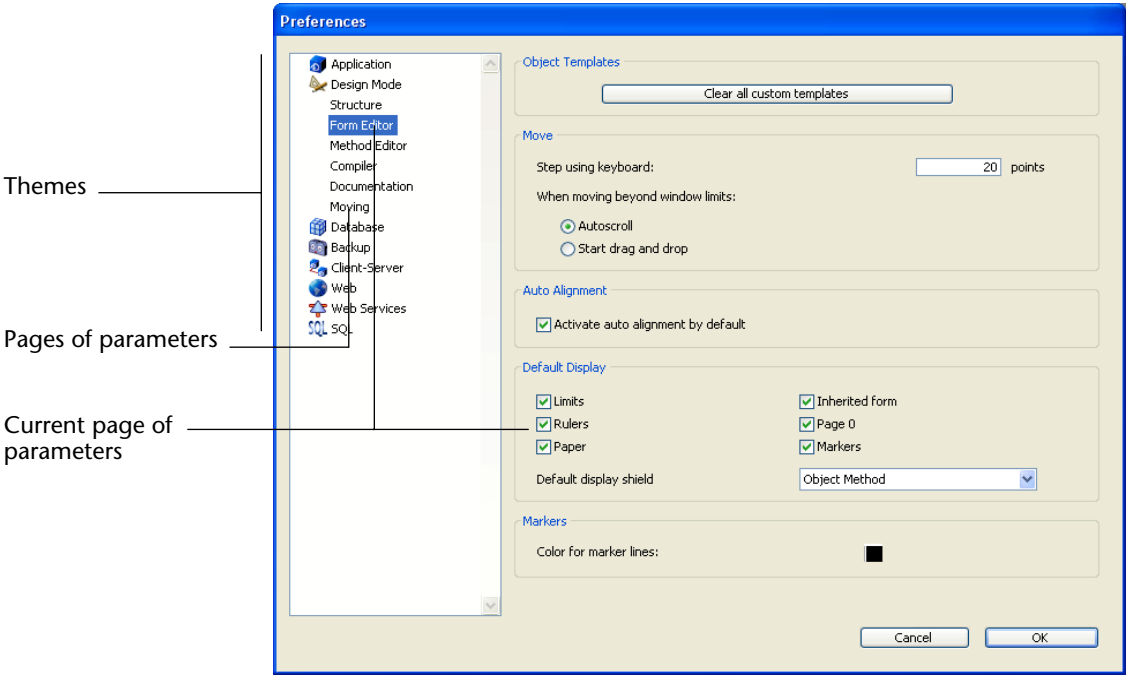


Note It is also possible to access the "Backup" theme of the Preferences directly from the Backup dialog box and the Compiler page from the compilation window.

The Preferences dialog box has several pages of parameters, grouped under various themes.

To display a page of parameters in the right-hand part of the window, you just need to expand the corresponding theme and select the name of the page in the list on the left-hand side.

You can use the **arrow keys** of the keyboard to expand/contract and select the themes. The **Tab** key lets you select the values of the page of parameters sequentially.

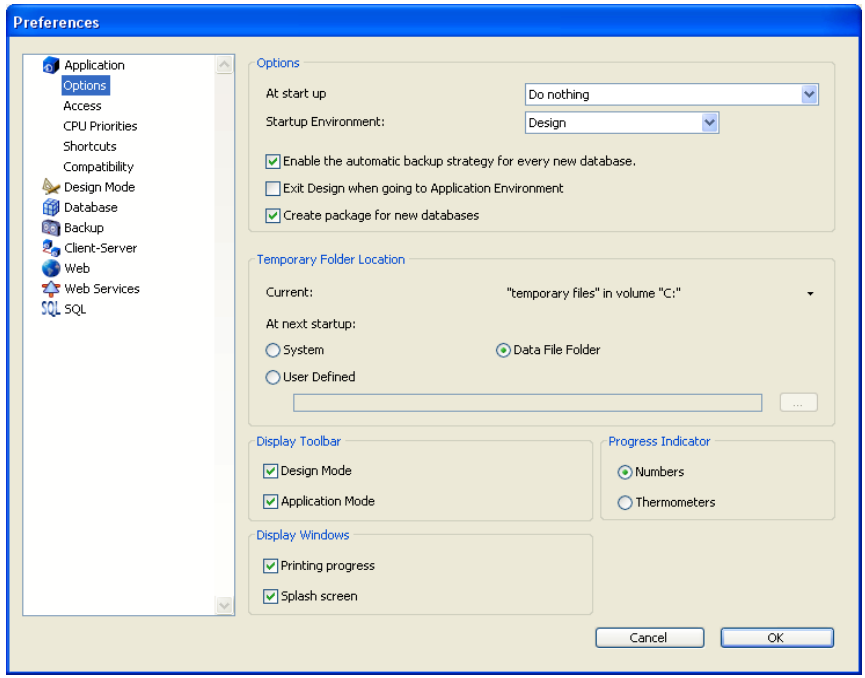


Application Theme

The pages of this theme let you configure certain aspects of 4D internal operation.

Options Page

The Options page lets you set various options concerning the startup and display of the 4D application:



Options

- **At startup:** Can be used to configure the default display provided by 4D on startup when the user launches only the application. The following options are available:
  - **Do nothing**: Only the application window appears, empty.
  - **Show open dialog** (default option): 4D displays the standard Open document dialog box.
  - **Open last used database**: 4D directly opens the last database used; no opening dialog box appears.

Note To force the display of the opening dialog box when the **Open last used database** option is selected, hold down the **Alt** (Windows) or **Option** (Mac OS) key while the database is being launched.

- **Startup Environment**: Select the environment in which you want the database to open by default: **Design** or **Application**. Unless you specify otherwise, by default 4D opens the database in the Design environment if there is not a password restricting its access.

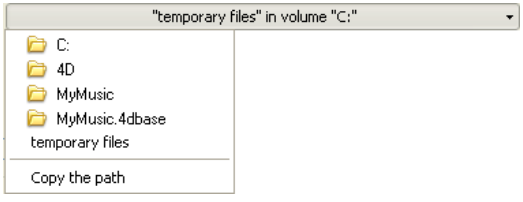
Note The choice between opening a database in compiled or interpreted mode is made in the opening dialog box (see the "Open Dialog Box Options" paragraph on page 40).

- **Enable the automatic backup strategy for every new database:** Implements a set of backup parameters in each new database (automatic backup, log file, etc.) ensuring minimum security. For more information about this option, please refer to the "Automatic Backup Strategy" paragraph on page 38.
- **Exit Design when going to Application Environment**: If this option is checked, when the user changes to the Application environment via the **Test Application** menu command, all the windows of the Design environment are closed. If this option is not checked (default value), the windows of the Design environment will remain visible in the background of the Application environment.
- **Create package for new databases**: When this option is checked, 4D databases are automatically created in a folder suffixed **.4dbase**. Thanks to this principle, under Mac OS the database folders appear as packages having specific properties. Under Windows, this has no particular impact. For more information, please refer to the ".4dbase Extension" paragraph on page 45.

Temporary Folder Location

This area lets you change the location of temporary files created while 4D is running. The temporary files folder is used by the application, when necessary, to temporarily save the data in memory to disk.

The current location of this folder is displayed in the "Current:" area. You can click in this area to show the pathname as a scrolldown list:



Three location options are provided:

- **System**: When this option is selected, the 4D temporary files are created in a folder placed at the location specified by Windows and/or Mac OS. You can find out the current location defined by your system using the Temporary folder command.
  The files are put into a subfolder whose name consists of the database name and a unique identifier.
- **Data File Folder** (default option): When this option is selected, the 4D temporary files are created in a folder named "temporary files" located at the same level as the data file of the database.
- **User Defined**: This option is used to set a custom location.

  If the location option is modified, it will be necessary to restart the database in order for the new option to be taken into account.
  4D checks whether the folder selected can be write-accessed. If this is not the case, the application will try other options until a valid folder is found.

Note This option is stored in the "extra properties" of the structure that is available when the structure definition is exported in XML (see the section "Exporting a Structure Definition" on page 331).

Progress Indicator

This option selects the progress indicators: **Numbers** or **Thermometers**.

You can monitor the progress of an operation such as indexing records through a numeric display or a graphic thermometer. The thermometer is slower, but easier to read. The numbers are faster, but they do not always refer to the number of records processed. For example, when 4D performs a sort, the numbers actually show the number of comparisons made.

Display Toolbar

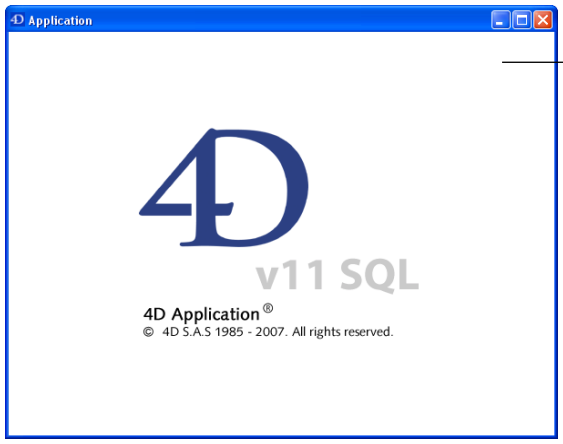
The options of this area control the default display of the general 4D toolbar in the Design and/or Application mode.
Keep in mind that this setting applies to each database opened with your 4D application.

Display Windows

The "Display Windows" area lets you set the default display of two types of windows in the application.

- **Printing progress**: This option lets you enable or disable the display of the printing progress dialog box when printing.

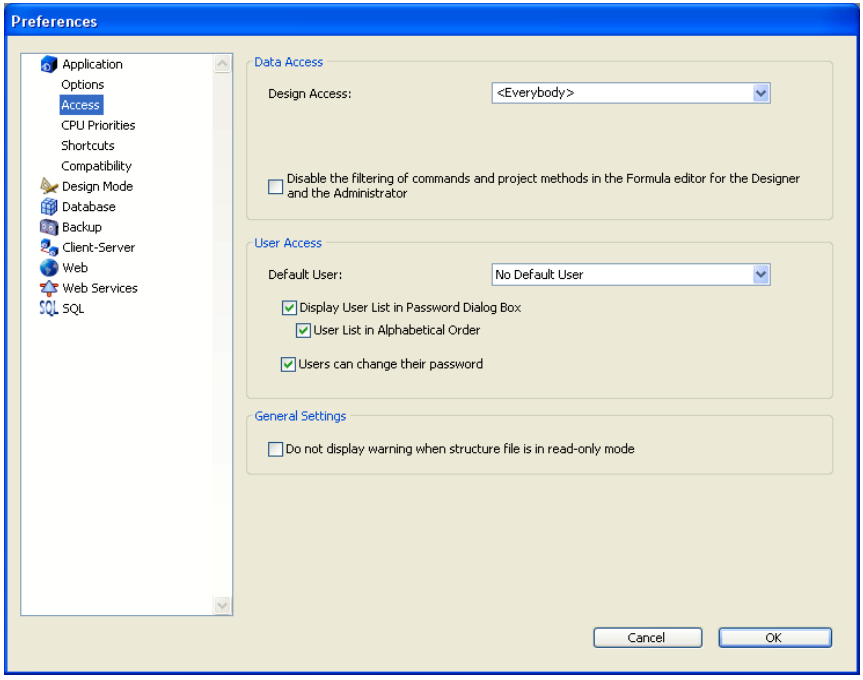
- **Splash screen**: When this option is deselected, the splash screen of the Application mode does not appear.



When you hide this window, it is up to you to manage the display of all your windows by programming, for example in the On Startup database method.

Access Page

The Access page lets you control access to the Design environment of 4D and to set the access conditions for users:



Data Access

- **Design Access**: Gives the specified group the ability to enter the Design environment of the database.

Notes
- Setting an access group in the Design environment also lets you deactivate the **Create table** option in the data import dialog box. For more information about this dialog box, refer to the "Importing Data from Files" paragraph on page 1001.
- The Designer and Administrator always have access to the Design environment, even if they are not explicitly part of the specified access group.

Note For more information about users and user groups, refer to the chapter "Users and Groups" on page 831.

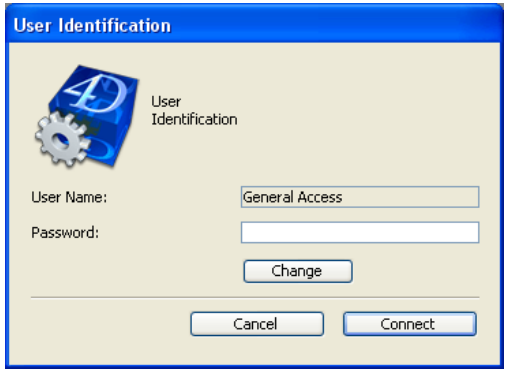
- **Disable the filtering of commands and project methods in the Formula editor for the Designer and the Administrator**: This option re-establishes full access for the Design and Administrator to 4D commands and to methods in the Formula editor.
For security reasons, 4D restricts access to these elements in standard editors (Formula editor, Quick Report editor, and so on): only certain 4D functions and project methods that are explicitly declared via the SET ALLOWED METHODS command can be used in these editors.
The **Disable the filtering...** option can be used to set up an unlimited access mode to commands and methods while remaining in control of the operations carried out. During the development phase, this mode can be used to freely test all the formulas, reports, and so on. During operation, it can be used to set up secure solutions that allow access to commands and methods on a temporary basis. This consists in changing the user (via the CHANGE CURRENT USER command) before calling a dialog box or starting a printing process that requires full access to the commands, then returning to the original user when the specific operation is completed.

Note If full access has been enabled via the **Always allow...** option described in the "Compatibility Page" paragraph on page 173, this option will have no effect.

This option is checked by default for databases created with 4D version 2004.4 and higher. In the case of converted databases, it is unchecked by default.

User Access

- **Default User**: When a Default User has been set, every user that opens the database or logs onto it has the same access privileges and restrictions defined for this Default User. A default user is not required to enter a user name. Moreover, if you have not associated a password with the Default User, the Password dialog box no longer appears and the database opens directly.
  This option simplifies access to the database while maintaining a complete data control system.
  - If you have associated a password with the Default User, a dialog box appears when the database is opened and the users must enter a password:



  - If you haven't associated a password with the Default User, the above dialog box will not appear.

Note You can "force" the display of the User Identification dialog box when the "Default User" mode is active, for instance in order to connect as Administrator or Designer. To do so, press the **Shift** key while opening the database or connecting to it.

- **Display User List in Password Dialog Box**: If this option is checked, users must choose their name from the list of users and enter their password in the User Identification dialog box. If it is not checked, users must enter both their name and password. For more information about the two versions of the password dialog box, see the section "Access System Overview" on page 832.
- **User List in Alphabetical Order** (only available if the previous option is checked): When this option is checked, the list of users in the password entry dialog box is sorted by alphabetical order.

- **The user can change their password**: When this option is checked, a **Change** button is displayed in the User Identification dialog box. This button lets the user access a dialog box that can be used to change their password (for more information about this dialog box, refer to the "Modification of Password by User" paragraph on page 857).
If desired, you can hide the **Change** button so that users cannot modify their passwords. To do so, simply deselect this option.

Note For compatibility reasons, this option is deselected by default in databases created with a version of 4D prior to 2004.

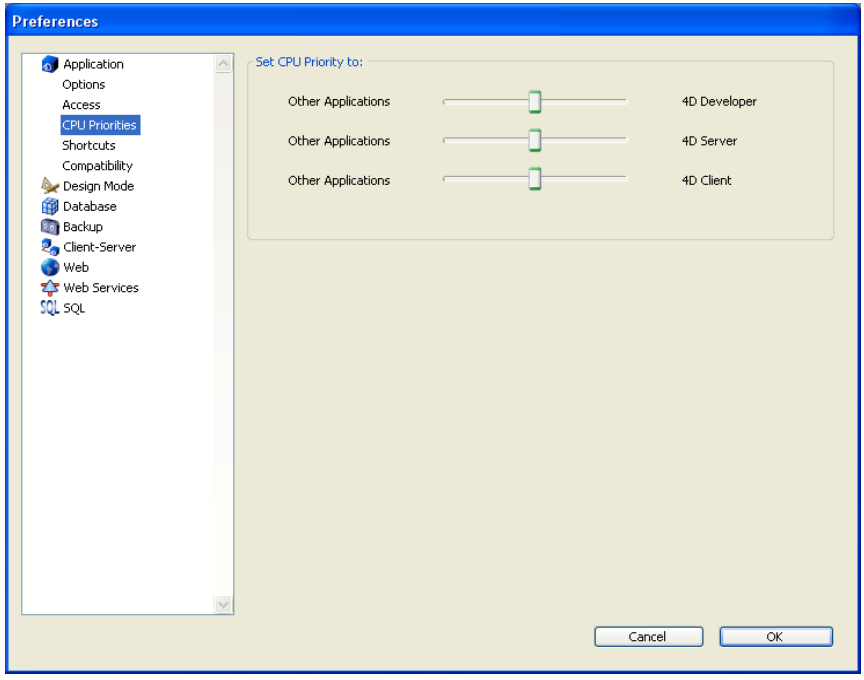
Do Not Display Warning When Structure File is in Read-only Mode

When you open a database whose structure file is in read-only mode, 4D displays a warning dialog box indicating this fact. If you open the database, any modifications made to the structure file will not be saved.
In certain cases, you might not want this dialog box to appear — for instance, in the case of consultation databases stored on CD-ROM. In this case, simply check the **Do not display warning when structure file is in read-only mode** option.

Notes
- For compatibility reasons, this option is checked by default for databases created with a version of 4D earlier than 2004.1.
- It is also possible to configure the application in the case of a data file being locked (see the "General Settings" paragraph on page 193).

CPU Priorities Page

The CPU Priorities page lets you optimize 4D application performances by modifying the frequency with which 4D applications must “switch back” to other applications.



In fact, 4D “bunks” with the machine’s operating system that hosts it as well as possibly hosting other applications. Since the architecture of systems on which 4D is run is not designed to multi-task, the system hands off to an application who then switches back to the system and so on. When an application switches back to the system, it indicates the time limit that it wishes to leave other applications.

Depending on your working environment and your specific needs, you can set the priority thermometer allocated to 4D with respect to other applications operating on the same machine. For a similar database, it is possible to set the priority for 4D Developer, 4D Server and 4D Client separately. These settings are stored in the structure file.

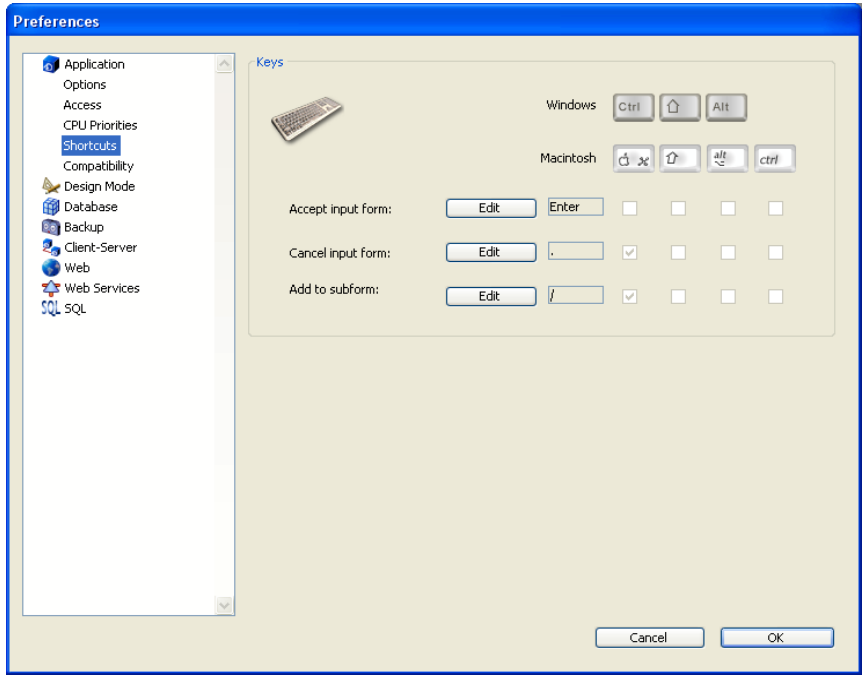
By default, the values correspond to “average” settings (cursor located in the center of the thermometer). The default values result in a machine time consumption equal to about 20%. In some cases this percentage may not be sufficient; if, for example, you attribute a dedicated machine to your 4D Server, it would be a pity to set aside 80% of machine time to a system that will do nothing else... In this case, it is most likely that raising the “4D Server” cursor priority will improve performance. Conversely, in other cases, it will be a good idea to decrease priority in order to force 4D to leave time for other applications.

Notes

- It is possible to know and set each setting through programming using the Get database parameter and SET DATABASE PARAMETER commands.
- Warning: Modifying the value of these cursors inappropriately can lead to the deterioration of application performances.

Shortcuts Page

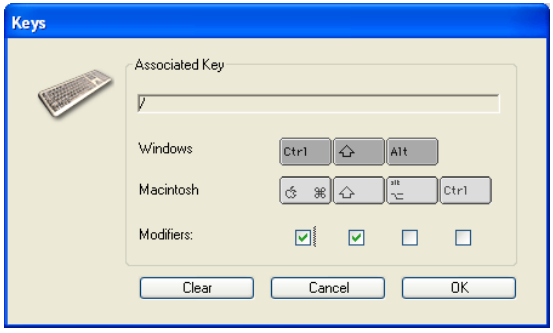
The Shortcuts page allows viewing and modifying default shortcuts for three basic 4D operations: Accept input form, Cancel input form and Add to subform.



These shortcuts are identical for both platforms. Icons in the form of keys indicate the corresponding Windows and Mac OS keys. The default shortcuts are as follows:

- Accept input form: **Enter**
- Cancel input form: **Esc**
- Add to subform: **Ctrl+Shift+/** (Windows) or **Command+Shift+/** (Mac OS).

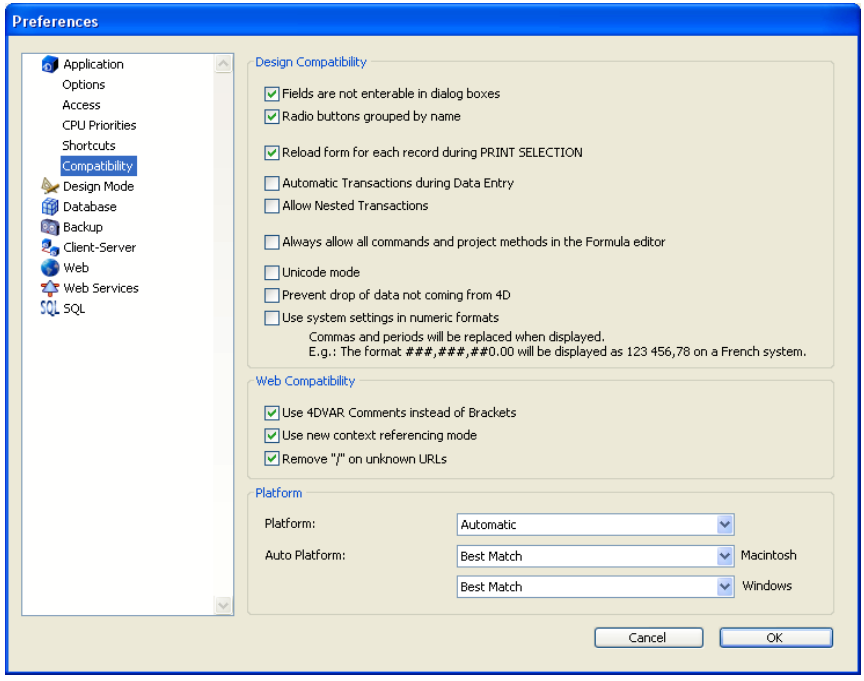
To change the shortcut of an operation, click the corresponding **Edit** button. The following dialog box appears:



To change the shortcut, type the new key combination on your keyboard and click **OK**.
If you prefer not to have a shortcut for an operation, click the **Clear** button.

Compatibility Page

The Compatibility page groups together parameters related to maintaining compatibility with previous versions of 4D. Keep in mind that the number of options displayed will depend on the version of 4D with which the original database was created: 2003.x, 2004.x or v11.



Design Compatibility

The options of this area concern the functioning of the 4D engine and the Design environment.

- **Fields are not enterable in dialog boxes**: In previous versions of 4D, it was not possible to enter values using fields in dialog boxes (displayed, for example, using the DIALOG command). This limitation has been removed since 4D 2004.
  You can still keep the previous behavior, especially if your database uses fields in dialog boxes to display data.
  By default, this option is checked for previous databases converted to version 2004 and is unchecked for databases created in version 2004.

- **Radio buttons grouped by name**: In previous versions of 4D, the coordinated[1] behavior of a group of radio buttons was obtained by giving the same first letter to the variables associated with the buttons (for example, *m_button1*, *m_button2*, *m_button3*, etc.).
  This was sometimes less than satisfactory and beginning with 4D 2004 it was changed as follows: to operate in a coordinated manner, a set of radio buttons must simply be grouped in the Form editor. For more information about this, refer to the section "Radio Buttons and Picture Radio Buttons" on page 574.
  This new mode is valid for radio buttons, 3D radio buttons and picture radio buttons.
  For compatibility reasons, the former mode is kept by default in converted databases. However, you can force the use of the new mode by deselecting this option.
  This option only appears for databases converted from former versions. It is checked by default. Databases created in version 2004 use the new mode.

- **Reload form for each record during PRINT SELECTION**: In previous versions of 4D, the form used during a print using the PRINT SELECTION command was reloaded for each record. This allowed automatically reinitializing all object settings that the developer might have changed using language in the On printing detail form event.
  In order to optimize performance, this mechanism was deleted beginning with 4D 2004. The 4D developer must now reinitialize the desired settings in the form method himself — this is identical to how list forms work with the On display detail form event. Nevertheless, you can keep the former mechanism using this option.
  This option only appears for databases converted from previous versions. It is checked by default. Databases created in version 2004 use the new mode.

- **Automatic Transactions during Data Entry**: Automatically starts a multi-transaction when an input form is first opened when the form has a subform. *This option is intended only for 4D First users. It should not be used for 4D applications, where it is preferable that the transactions be handled by the developer.*

1. Coordinated behavior means that only one button in the group can be selected at a time.

- **Allow Nested Transactions**: Enables support of multi-level transactions. Beginning with v11, 4D accepts nested transactions on an unlimited number of levels. Since this new operation can lead to malfunctioning in databases developed with former versions of 4D, it is disabled by default in converted databases (transactions remain limited to a single level). If you want to take advantage of transactions on several levels in a converted database, you must check this option.
By default, this option is not checked. It is specific to each database.

Note The **Allow Nested Transactions** option has no effect on transactions carried out in the SQL engine of 4D v11. SQL transactions are always multi-level.

- **Always allow all commands and project methods in the Formula editor**: This option can be used to disable the limiting of user actions in the standard editors of 4D (Formula editor, Quick Report editor, and so on). When this option is checked, users have access to all the 4D commands and plug-ins as well as all the project methods (except for invisible project methods) in the standard editors. This operation is the same as that of 4D versions prior to 2004.

Notes • This option takes priority over the SET ALLOWED METHODS command. When it is checked, this command has no effect.
• It is possible to partially disable the limiting of commands and methods using the **Disable the filtering of commands...** option (see the "Access Page" paragraph on page 166).

This option is checked by default for databases converted from a 4D version prior to version 2004. In all other cases, it is unchecked by default.

- **Unicode mode**: Used to enable or disable the Unicode mode for the current database. In Unicode mode, the database engine, the language and the menus handle Unicode character strings natively. In non-Unicode mode (compatibility mode), the ASCII character set is used.
This option allows the compatibility of converted databases to be maintained. It is checked by default for databases created with 4D v11 and higher, and unchecked in converted databases.

Notes • This option is specific to each database. It is thus possible to have a Unicode database coexist with non-Unicode components (or vice versa) in interpreted mode.
• It is also possible to configure the Unicode mode using the Get database parameter and SET DATABASE PARAMETER BASE commands.

The specific characteristics of Unicode support in 4D are detailed in the *Language Reference* manual.

- **Prevent drop of data not coming from 4D**: Starting with v11, 4D allows drag and drop of selections, objects and/or files external to 4D, like picture files for example.
This possibility must be supported by the database code. In databases converted from previous versions of 4D, this possibility may lead to malfunctioning if the existing code is not adapted accordingly.
The **Prevent drop of data not coming from 4D** option can be used to anticipate this possible malfunctioning. When it is checked, the dropping of external objects is refused in 4D forms. Note that inserting external objects is still possible in objects having the **Automatic Drop** option, when the application can interpret the data being dropped (text or picture). For more information, refer to the "Automatic Drag and Drop" paragraph on page 543.
This option is checked by default in converted databases and unchecked by default for databases created in version 11.

- **Use system settings in numeric formats**: 4D v11 uses regional system parameters for numeric display formats (see "Number Field Formats" on page 525). 4D automatically replaces the "," and "." characters in numeric display formats by, respectively, the thousand separator and the decimal separator defined in the operating system. The period and comma are thus considered as *placeholder characters*, following the example of 0 or #. In previous versions of 4D, numeric display formats do not take the regional parameters of the system into account. For example, the "###,##0.00" format is a valid format for an American system. However, when it is applied to a numeric value displayed on a French or Swiss system, the result is incorrect.
In converted database, for the sake of compatibility, this new mechanism is not activated. To take advantage of it, you must check the "**Use system settings in numeric formats**" option. This option only appears in databases created with 4D v11 and higher.

Web Compatibility

The options of this area concern the operation of the 4D Web server.

- **Use 4DVAR Comments instead of Brackets**: This option sets the notation to use when inserting 4D expressions on static pages.
  - When the box is checked, the syntax you need to use is the standard HTML notation (<!--4DVAR MYVAR-->)[1].
  - When the option is not checked (default value), the syntax you need to use is the notation with square brackets ([MYVAR]) — which is a proprietary solution used in former versions of the 4D Web server.
- **Use new context referencing mode**: When this option is selected (default value), the 4D Web server places the context number in the basic URL of the HTML documents being sent.
  With the former system (option not checked), the 4D Web server sends the context number for each item of a page to the browser, which slows down processing. This option may nevertheless be left unchecked for compatibility reasons. Keep in mind that you must restart the database after modifying this option in order for the new operation to become effective.
- **Remove "/" on unknown URLs**: In previous versions of 4D, unknown URLs (URLs that do not correspond to an existing page nor to a 4D special URL) were returned in the On Web Authentication and On Web Connection ($1) database methods and did not begin with the "/" character. This specificity was removed in 4D 2004.
  However, if you implemented algorithms based on this special case and wish to keep the previous behavior, just check the **Remove "/" on unknown URLs** option.
  By default, this option is checked in converted databases and unchecked in new databases.

1. A space character must be inserted between 4DVAR and the variable name.

Platform

This area only appears in converted databases. It gives access to former parameters concerning the platform interface, i.e. the appearance of database forms. In previous versions of 4D, it was possible to "force" this appearance at the level of the database, forms and/or objects — each level inheriting from the previous by default. This mechanism has been kept for compatibility reasons.

In databases created with 4D version 2004 or higher, the interface is automatically chosen by 4D according to the current platform. The forms and objects used will be built with a native design. The interface can be configured only at the form or object level and only two options are available: **System** and **Printing**. For more information about these options, refer to the "Platform" paragraph on page 396.

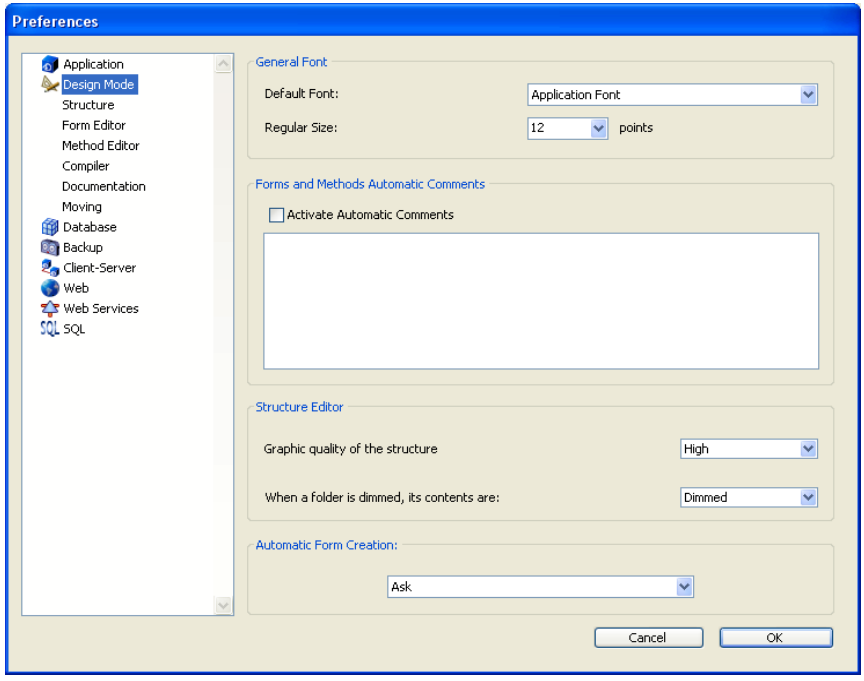
In the application Preferences, only the **System** option appears in the "Platform" pop-up menu (as well as former options). It is strongly recommended that you select this option in order to implement native design for converted databases. In this case, the two other pop-up menus are disabled. Keep in mind that you might need to modify the "Platform" property of your forms in order to take advantage of this mechanism.

Design Mode Theme

The pages of this theme let you configure various preferences relating to the interface and to the operation of the windows and editors of the 4D development environment.

Structure Page

This page lets you modify the general font (used mainly in the Design mode windows), set automatic comments and configure the Structure editor.



General Font

The **Default Font** and **Regular Size** menus can be used to set the font and size of characters used in certain windows of the 4D application, more particularly:

- The Structure editor
- The preview area of the Explorer,
- The Runtime Explorer,
- The Form editor ruler,
- The List of tables window,
- The help tips.

Forms and Methods Automatic Comments

This area lets you activate and set the automatic comment system in your database.

When you check the **Activate Automatic Comments** option and enter text in the entry area below it, this text will be automatically associated with each form or method that is created or modified in the database. Comments can be viewed in the Explorer. For more information, refer to the "Inserting Automatic Comments" paragraph on page 134.

Structure Editor

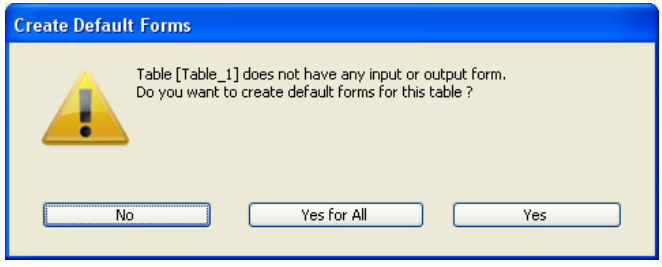
This area lets you set options relating to the display of the 4D Structure editor window:

- **Graphic quality of the structure**: This option can be used to vary the level of graphic detail in the Structure editor. By default, the quality is set to **High**. You can select **Standard** quality in order to give priority to display speed. The effect of this setting is mainly perceptible when using the zoom function (see the "Zoom" paragraph on page 251).
- **When a folder is dimmed, its contents are**: This option lets you set the appearance of dimmed tables in the structure editor, when you carry out selections by folder (see the "Highlight/Dim Tables by Folder" paragraph on page 252). The possible options are **Dimmed** (a shadow replaces the table image) and **Invisible** (the table disappears completely).

Note It is necessary to close and reopen the Structure editor window in order for this preference to be taken into account.

Automatic Form Creation

This menu lets you set the behavior of 4D when you create a table in the Structure editor and then open, for example, the records display window. By default, 4D tells you that no form has been created for the new table and then gives you the option of creating default input and output forms automatically:

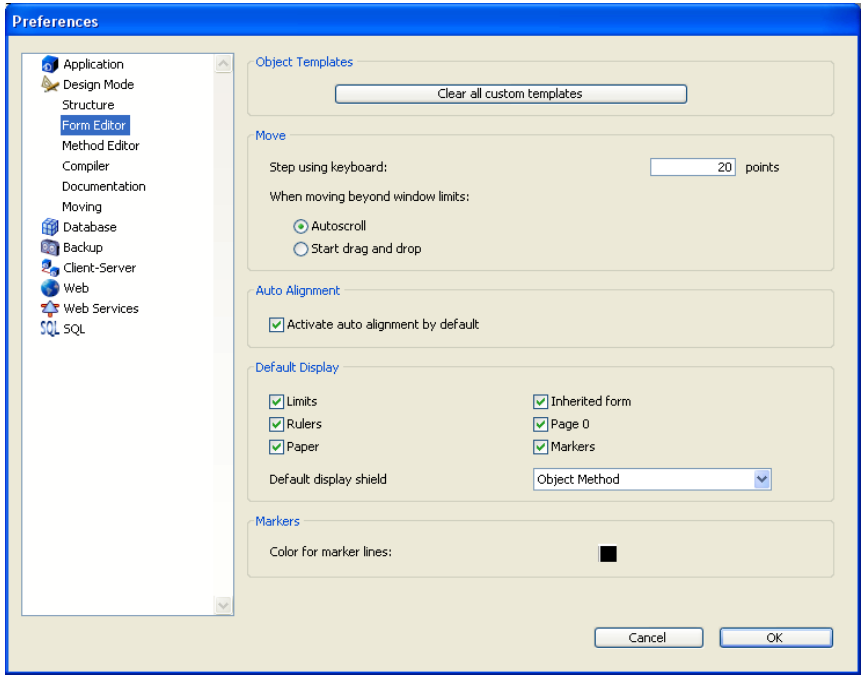


There are three options available in the menu:

- **Never**: The alert dialog box doesn't appear and no default form is created.
- **Ask**: The alert dialog box appears when no form for the table has been created (default option).
- **Always Yes for All**: The alert dialog box doesn't appear, but default forms are created for all the tables automatically.

Form Editor Page

This page lets you set the default operation of the 4D Form editor.



Object Templates

4D lets you set a custom form object as a "template," which will be used subsequently to create all the objects of the same type. For more information about this, refer to the "Setting an Object Template" paragraph on page 430.

The **Clear all custom templates** button can be used to reset all the object templates to their default values.

Move

This area is used to set parameters for moving objects using the keyboard or the mouse in the Form editor.

- **Step using keyboard**: This option allows setting the value (in points) of the step used for moving or resizing an object using the keyboard and the **Shift** key.
- **When moving beyond window limits**: This option allows setting the behavior of the Form editor when moving an object using the mouse beyond window limits.
  - **Autoscroll**: When this option is checked, this action causes the scroll of the form in the window, as if you clicked on the scroll bars. This behavior is useful for moving objects in large forms.
  - **Start drag and drop**: When this option is checked, this action is interpreted as a drag and drop. The form window is not modified and the moved object can be dropped in another window (if its contents are compatible), for example, in another form. This behavior is useful for recycling objects among several forms or using object libraries (see the "Using Object Libraries" paragraph on page 461).

  You can configure this option depending on your work habits and development needs. By default, the autoscroll option (traditional 4D behavior) is selected.

Auto Alignment

The **Activate auto alignment by default** option allows activating auto alignment by default in each new window of the Form editor. It is possible to modify this option individually in each window (please refer to the "Using the Magnetic Grid" paragraph on page 423).

Default Display

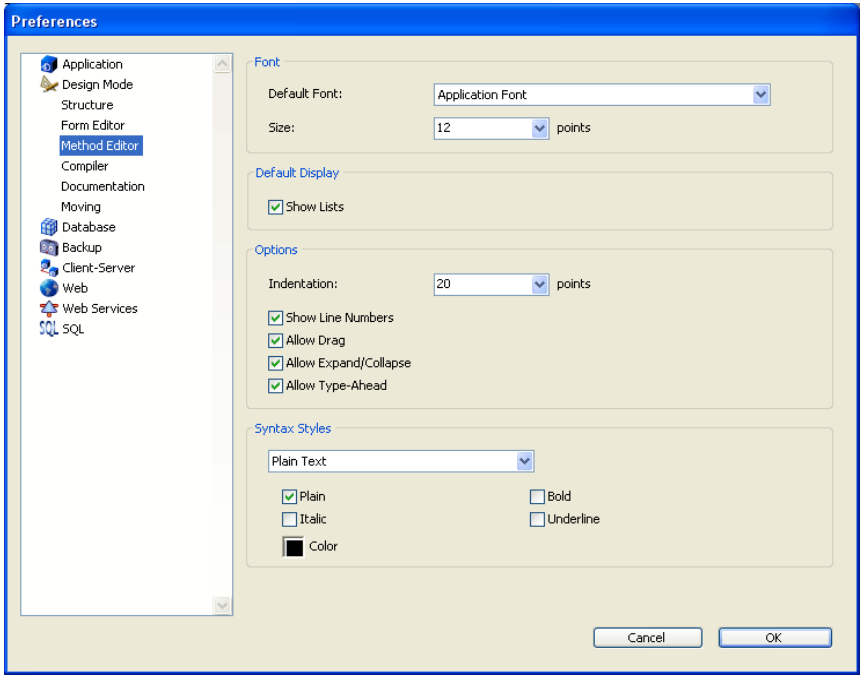
These options allow setting the items displayed or hidden by default in each new window of the Form editor. It is possible to modify the display of each window individually using the **Display** hierarchical menu of the Form editor.
You can also set the **Default display shield**. For more information on shields, please refer to the "Using Shields" paragraph on page 434.

Markers

The **Color for marker lines** option allows modifying the color of the marker lines used in the Form editor to define the different areas (header, breaks, detail and footer, etc.). For more information about markers, refer to the "Output Control Lines" paragraph on page 642.

Method Editor Page

This page lets you set the interface, default display and operating options of the Method editor:



For more information about the Method editor, refer to the "Method Editor Window" paragraph on page 781.

Font

- **Default Font** and **Size**: These menus let you set the font and font size to be used in the entry area of the Method editor.

Note You can set the font style for each different type of object. For more information, refer to the "Style of Syntax Elements" paragraph on page 185.

Default Display

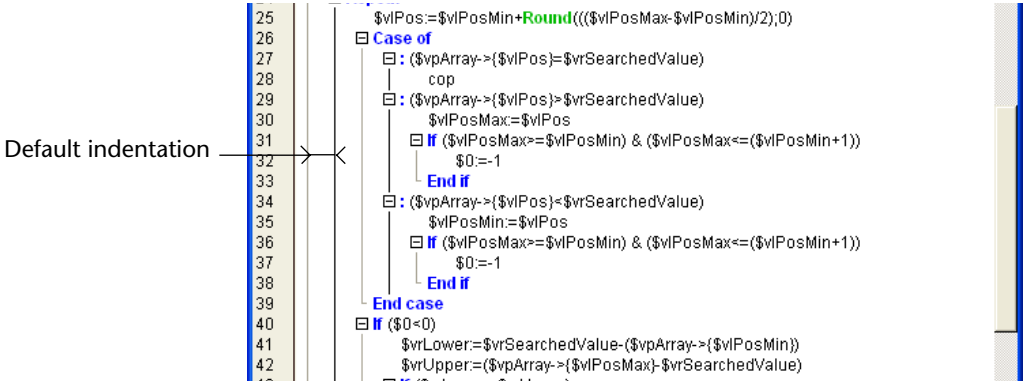
- **Show Lists**: This option lets you choose whether or not to show the lists of objects (Commands, Tables and fields, etc.) by default when the Method editor window is opened. By default, this option is checked and the lists are displayed.

Note It is possible to show or hide the lists for the current window directly from the Method editor. Refer to the "Method Editor Window" paragraph on page 781.

Options

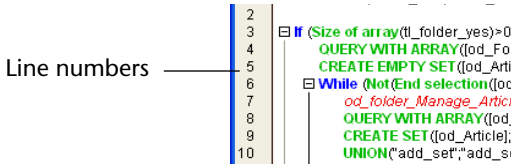
- **Indentation**: This option lets you change the indentation value for the code in the Method editor. The width must be specified in points (20 by default).

  4D code is automatically indented in order to reveal its structure:



  Modifying this default value can be useful if your methods contain complex algorithms with many levels of embedding. Narrower indentation can be used in order to limit horizontal scrolling.

- **Show Line Numbers**: This option lets you display the line numbers in each window of the Method editor:

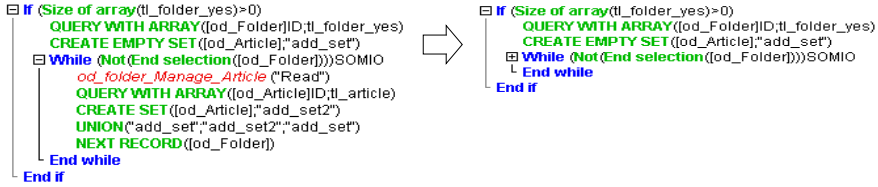


Note It is possible to show/hide the line numbers for the current window directly from the Method editor. Refer to the "Display of Line Numbers" paragraph on page 789.

- **Allow Drag**: This option (checked by default) lets you enable/disable the drag-and-drop mechanism inside a method or between two methods.

Note This option does not affect the drag-and-drop mechanism of the 4D Explorer.

- **Allow Expand/Collapse**: This option (checked by default) lets you enable/disable the display of 4D code located inside loops and conditions in the form of hierarchical lists that can be expanded and collapsed:



When this option is not checked, the expand/collapse icons disappear and the code is displayed in expanded form.

- **Allow Type-Ahead**: This option (checked by default) lets you enable/disable the type-ahead function provided in the Method editor of 4D. For more information about these functions, refer to the "Type-ahead Function" paragraph on page 800.

Style of Syntax Elements The options of this area let you assign a specific style and color to each type of element of the 4D language (fields, tables, variables, parameters, SQL, etc.). The combination of different colors and styles for the method elements can prove to be extremely useful for code maintenance.

To set the graphic attributes of a syntax element, select it in the pop-up menu then choose the desired style options. You can apply several different attributes to the same element.
To set a color, click on the selection area in order to make the color palette appear.
You can set the attributes for several elements one after the other — it is not necessary to click **OK** after each modification.

Notes
- The "Plain text" element type indicates all the texts not belonging to any other defined type (i.e., symbols, punctuation, literal constants, etc.).
- The "Keywords" element type indicates the programming structures (If/End if, Case of/End case, etc.) that are accessible via Macros.

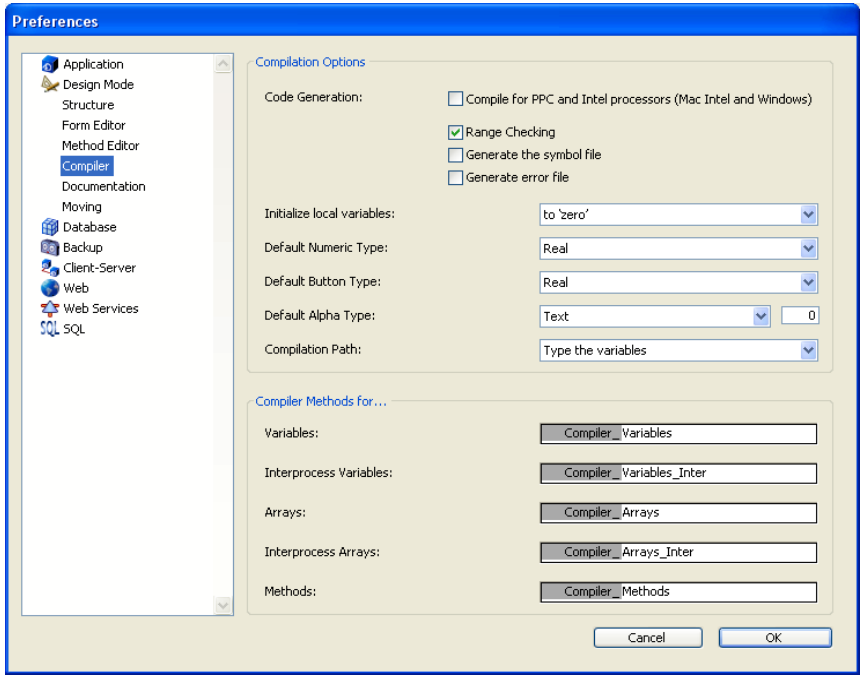
The available styles and their default use are the following:

- **Normal**
- **Bold** (associated by default with 4D commands, keywords and plug-in commands).
- **Italic** (associated by default with plug-in commands and methods)
- **Underline** (associated by default with predefined constants)
- **Color** (colors are associated by default with the main elements).

Compiler Page

This page lets you set parameters relating to database compilation. For more information about this, refer to the chapter “Compilation” on page 1229.

The parameters set will be applied to all the databases opened with the current 4D application.



Compilation Options

This area groups the generic options used during the compilation process.

- **Compile for PPC and Intel processors (Mac Intel and Windows)**: By default, when this option is not checked, 4D generates the compiled code corresponding to the processor of the platform on which the application is running.
  If you check this option, 4D will generate both PowerPC (Mac OS "standard") and Intel (Mac Intel and PC Windows) compiled code, regardless of the running platform.

- **Range Checking**: Used to activate/deactivate range checking. Range checking is an additional verification that checks the code in situ and according to the state of database objects at a given moment.
  For more information about range checking, refer to the "Range Checking" paragraph on page 1248.

- **Generate the Symbol File**: This option is used to generate an ASCII type file containing the list of variables along with their type and the method from which their type has been inferred. The symbol file also contains the list of your methods and functions along with the type of their parameters and the type of result, if any.
  The file is placed in the folder containing the database structure and is named:
  - Under Windows, DatabaseName.sym,
  - Under Mac OS, DatabaseName.symb.

  For more information about the symbol file, refer to the "Symbol File" paragraph on page 1243.

- **Generate error file**: This option is used to generate the error file at the time of syntax checking. It lists general errors as well as errors linked to a specific line, and warnings.
  Any errors detected by the compiler are automatically accessible in the **Method** menu of 4D. However, having an error file that can be transmitted from one machine to another can be useful, particularly in a situation where several different developers are working together in a client-server environment.
  The error file is generated in XML format in order to facilitate automatic parsing of its contents. It also allows the creation of customized error display interfaces.

The error file is automatically named DatabaseName_errors.xml and is created next to the structure file of the database,

For more information about the error file, refer to the "Error File" paragraph on page 1245.

- **Initialize Local Variables**: This option is used to set the local variable initialization mode at the beginning of methods:
  - **to 'zero'**: Variables are reset to zero by default (empty string for character strings, 0 for numbers...).
  - **to a random value**: The compiler assigns a random value, always the same, (1919382119 for longints, "rgrg" for character strings, True for Booleans...) to variables. This option enables you to pinpoint local variables that you have forgotten to initialize.
  - **no**: The compiler does not initialize the variables. In this way, you gain time during database execution, provided that your initialization was correct.
- **Default Numeric Type**: This option is used to force numeric typing in an unambiguous manner, either in real or longint. It has no priority over any directives that may have been placed in your database. You can optimize the running of your database by choosing the Longint type.
- **Default Button Type**: This option is used to force button typing in an unambiguous manner, either in real or longint. It has no priority over any directives that may have been placed in your database. It concerns standard buttons as well as the following objects: check boxes, 3D check boxes, highlight buttons, invisible buttons, 3D buttons, picture buttons, button grids, radio buttons, 3D radio buttons, picture radio buttons, picture pop-up menus, hierarchical pop-up menus and pop-up/drop-down lists.
- **Default Alpha Type**: When the database is running in non-Unicode mode, this option is used to force character string typing in an unambiguous manner, either in text or fixed string. It has no priority over any directives that may have been placed in your database.
  If you choose to give character strings the **Fixed string** type by default, the enterable area allows you to indicate the length of these strings to the compiler (you must enter a value between 2 and 80). Choosing the Fixed string type lets you optimize the running of your databases.

When the database is running in Unicode mode, this option does not appear. In Unicode mode, the Text type is automatically used.

- **Compilation Path**: This option is used to set the number of passes performed by the compiler and thus the duration of compilation.
  - **Type the variables**: Passes by all the stages that make compilation possible.
  - **Process and interprocess are typed**: The pass for typing process and interprocess variables is not carried out. This option can be used when you have already carried out the typing of all your process and interprocess variables either yourself or using the function for the automatic generation of compiler methods.
  - **All variables are typed**: The pass for typing local, process and interprocess variables is not carried out. Use this option when you are certain that all the process, interprocess and local variables have been clearly typed.

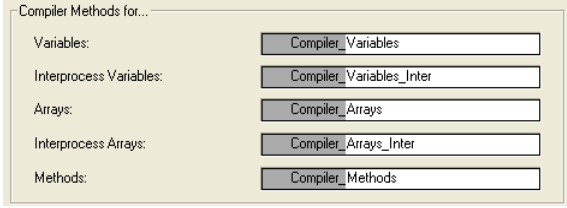
Compiler Methods

This area lets you rename the Compiler methods that are generated automatically by the compiler. These methods group together all the variable typing declarations, process and interprocess arrays, as well as the local variable declaration methods. These methods are generated using the compilation window. For more information, refer to the "Compiler Window" paragraph on page 1236.

Up to 5 compiler methods may be generated; a compiler method is only generated if the database contains the corresponding items:

- **Variables**: Groups together process variable declarations;
- **Variables_Inter**: Groups together interprocess variable declarations;
- **Arrays**: Groups together process array declarations;
- **Arrays_Inter**: Groups together interprocess array declarations;
- **Methods**: Groups together local variable declarations designating method parameters (for instance, C_INTEGER(mymethod;$1)).

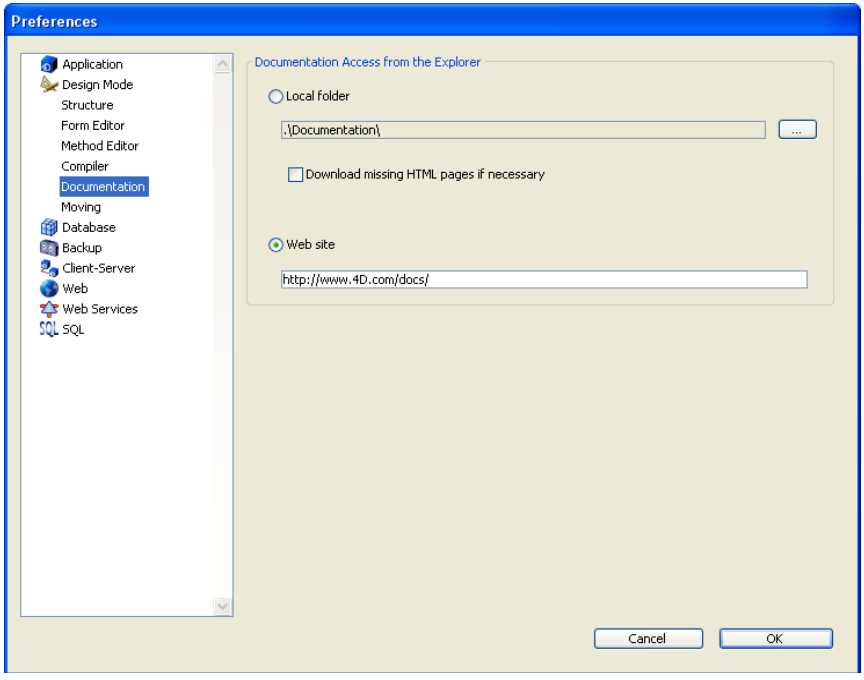
You can rename each of these methods in the corresponding areas:



Nevertheless, they will always be preceded by the label "*Compiler_*" (non-modifiable). The name of each method (prefix included) must be unique and no longer than 31 characters. Extended characters (accented characters, typographical symbols, etc.) and spaces are not allowed.

Documentation Page

This page lets you set the access to on-line documentation from the Explorer of 4D:



Note For more information about access to the on-line documentation from the Explorer, refer to the "Commands Page" paragraph on page 123.

- **Local folder/ Web site**: This option is used to indicate the location from which HTML documentation pages are loaded:
  - When the user double-clicks on a command in the Explorer (Windows and Mac OS);
  - When a command is selected in the Explorer (Mac OS only).

  If you select the **Local folder** option, 4D will look for the HTML pages in the specified folder. By default, this location is the Documentation folder located next to the 4D application (under Mac OS, next to the 4D software package).
  The indicated pathname is relative to the 4D application. You can modify this location as desired. The HTML documentation can be located on another volume, a CD-Rom, etc. To designate another location, click on the **[...]** button located next to the entry area and choose a documentation root folder (folder containing the 4DDOCUS.HTM, 4DDOCFR.HTM and 4DDOCGM.HTM files).

  If you select the **Web site** option, 4D will look for the HTML pages at the specified URL. The default URL corresponds to the standard 4D documentation URL on the 4D, Inc. Internet site. This URL can be modified as desired.

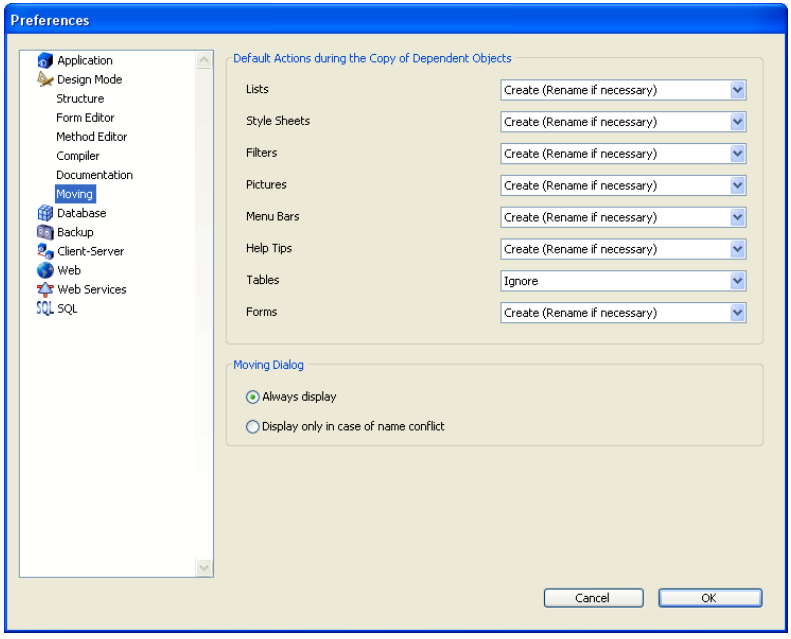
- **Download missing HTML pages if necessary**: This option is used to activate (or deactivate) automatic downloading of the HTML documentation pages from the Internet when they are not found in the "Local folder." This enables you to automatically integrate new 4D commands into your documentation or to fill the local folder gradually according to your needs.
  By default, this option is not checked: when you have chosen to load the documentation from the **Local folder**, if the page corresponding to a 4D command is not found in this folder, 4D displays an error page.

  However, if the **Download missing HTML pages if necessary** option is checked, 4D will download the page concerning the command from the URL specified in the "Web Site" area. Once it is downloaded, the page is stored in the local folder.

4D Server The settings for access to on-line documentation are specific to each client machine.

Moving Page

This page allows you to pre-configure the moving of objects in the database in Design mode:



These preferences will be applied to the dropping/pasting of objects into the database, when the database is used as a destination database.

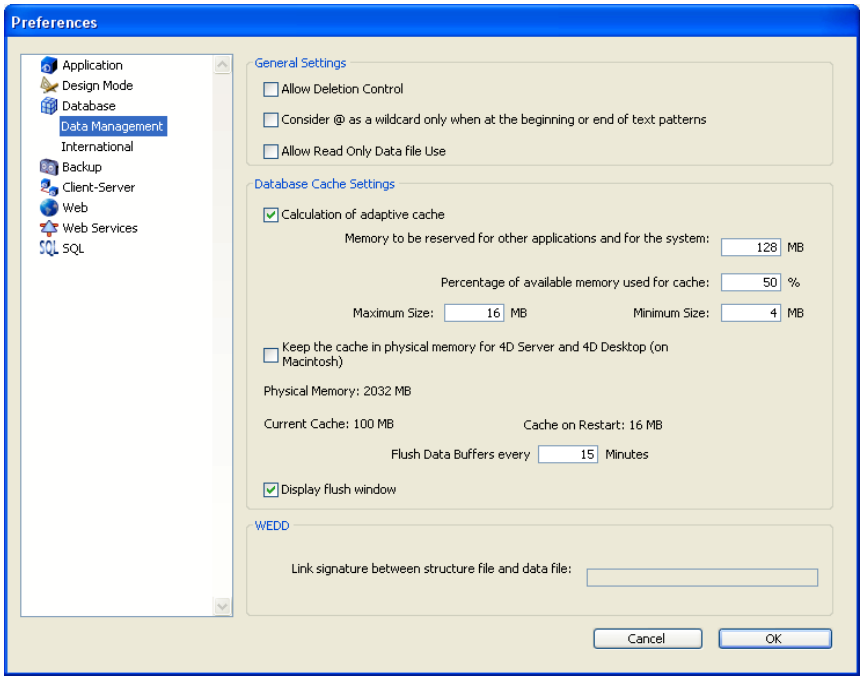
The options of this page are detailed in the "Moving Preferences" paragraph on page 157.

Database Theme

The pages of this theme let you set the preferences concerning the management and processing of the data of your database.

Data Management Page

This page lets you set the parameters related to data processing and to configure the cache memory for the database.



General Settings

You can use this area to set various options related to data integrity.

- **Allow Deletion Control**: This option allows you to use the Deletion control options in the Relation Properties window for each relation you define. If the **Allow Deletion Control** check box is not selected, the Deletion control options are disabled. For more information, see the "Deletion Control" paragraph on page 318.

- **Consider @ as a wildcard only when at the beginning or end of text patterns**: This option allows you to set how the *at* sign "@" will be interpreted when used in a query or a character string comparison, when it is located in a word.
  When this option is not checked (default value), the *at* sign is used as the wildcard character, in other words, it replaces any character (see the "Wildcard Character (@)" paragraph on page 957).
  When the option is checked, the *at* sign is regarded as a simple character if it is located within a word. This setting is especially useful when searching for E-mail addresses, where the @ sign is used internally.

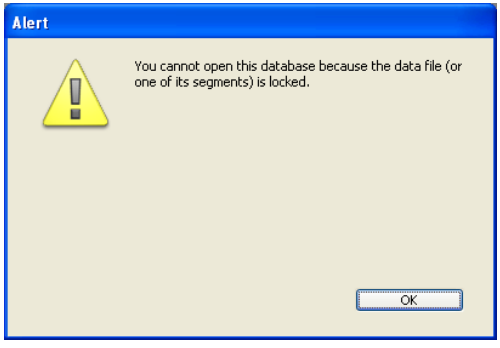
This option has an influence on searches, sorts, string comparisons, as well as on data stored in tables and data found in memory, like arrays. Fields and variables of the alpha (indexed or not) and text type are concerned by how the @ character is interpreted in searches and sorts.

Notes

- For searches, it is important to note that if the search criteria **begins** or **ends** with @, the "@" character will be treated as a wildcard. Only if the "@" character is placed **in the middle of** a word (for example: bill@cgi.com) will 4D treat it differently.
- This option can also have an influence on the behavior of the commands in the "Object Properties" theme that accept the wildcard character ("@") in the object parameter. Please refer to the *4D Language Reference* manual.
- For security reasons, only the Administrator or Designer of the database can modify this parameter.

If you modify this setting, you have to quit and reopen the database to make the change effective. Once the database is reopened, all of the database's indexes are automatically re-indexed.

- **Allow Read Only Data file Use**: This option allows configuration of the application operation when opening a locked data file at the operating system level.
4D includes a mechanism that automatically prevents the opening of a database when its data file, or one of its segments, is locked.
In this case, when this detection option is activated, 4D displays a warning message and does not open the database:



Unless this option is checked, it is not possible to open a database when its data file is locked (default operation for 4D databases). Note that this option is only applied to the open database and not to all the databases opened with the 4D application.

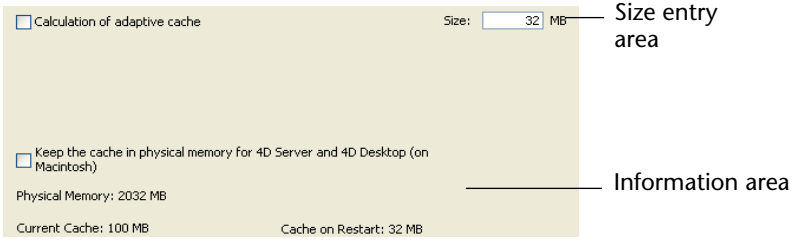
- **About locked files**
  Locked files may be read but their contents cannot be modified. For example, files are locked when they are stored on a non-rewritable support (CD-Rom type) or when they are recopied from this type of support. 4D can work in a transparent manner with locked data files, which allows, more particularly, the use of databases stored on CD-Rom.
  However, this operation runs the risk of inadvertent use of a locked data file in which modifications will not be saved. This is why by default 4D does not allow databases with a locked data file to be opened.

Note It is also possible to configure the database in the case of a locked structure file (see the "Do Not Display Warning When Structure File is in Read-only Mode" paragraph on page 169).

Database Cache Settings This area lets you configure the cache memory for the application data.

There are two modes of cache calculation: **"forced" mode** and **adaptive mode** (recommended and used by default).

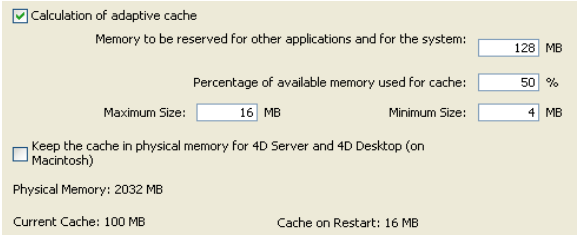
- **"Forced" mode** (**Calculation of adaptive cache** option not checked): In this mode, you set the size of the memory cache for the database yourself. 4D then displays an entry area that allows setting the memory cache to use as well as information related to the physical memory (RAM available on the machine), the current cache and cache after restart (taking your changes into account):



The size of the memory cache that you enter will be reserved for the 4D database, regardless of the state of machine resources.

This setting can be used in certain specific configurations, or when the database is designed to be used on dissimilar systems in terms of memory. In most cases, the adaptive cache offers better performance.

- **Adaptive mode** (**Calculation of adaptive cache** option checked): In this mode, management of the memory cache is done dynamically by the system — respecting limits that you set. This allows configuration of a high performance memory cache adapted to most configurations. 4D displays several additional entry areas:



The size of the memory cache is then calculated dynamically depending on set parameters. The values offered by default correspond to standard 4D usage.

- **Memory to be reserved for other applications and for the system**: Portion of the RAM memory to reserve for the System and other applications. This value is increased for optimization when other applications are running on the same machine as 4D.
- **Percentage of available memory used for cache**: Percentage of the remaining memory allocated to the cache by default.
  To obtain the size allocated by default to the cache, simply perform the following calculation: (Physical memory – Physical memory to be reserved) X Percentage of the memory used for the cache.

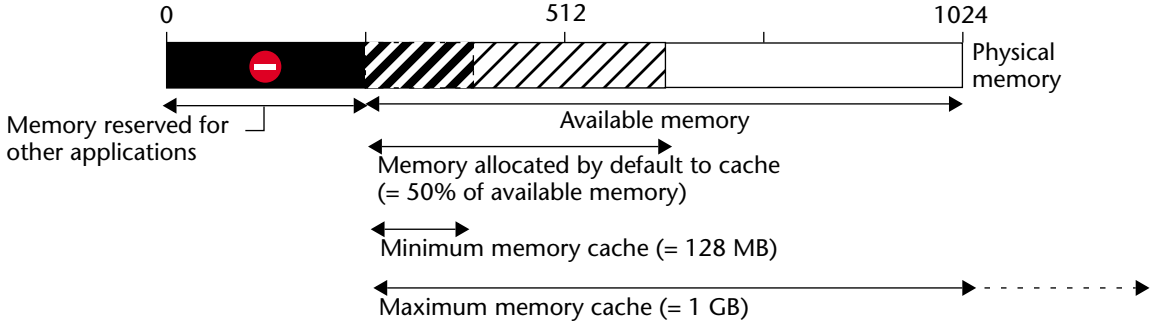
In the adaptive mode, the size of the memory cache varies dynamically depending on the needs of the application and the system. You can set limits using the following two options:

- **Maximum Size**: Maximum amount of memory that can be used by the cache. This value cannot be greater than 2000 MB (2 GB).
- **Minimum Size**: Minimum amount of memory that must be reserved for the cache. This value cannot be less than 4 MB.

Note Setting limits is particularly useful for databases that are distributed on machines for which you do not know the memory configuration *a priori*. In this case, the limits set let you guarantee a minimum performance in all cases.

The following diagram illustrates this behavior:

Memory cache calculation example
Physical memory to reserve = 256 MB
Percentage of the available memory used for the cache = 50%
Maximum size = 1 GB Minimum size = 128 MB



Examples of Configurations in Adaptive Mode

The following examples illustrate the advantages of having an adaptive cache in various software and memory configurations.

- Computer with 768 MB of RAM running a single 4D Server.
  In this case, it is worthwhile to allocate it with the maximum memory cache:
  - Physical memory to reserve = 256 MB.
  - Percentage of the available memory used for the cache = 100% since there is only one 4D Server application.
  - Minimum size = 10 MB and Maximum size = 1GB (in this configuration, these values are insignificant).

  Default cache size: (768 - 256) x 100% = 512 MB.

- Same configuration as before but with three 4D Server applications running on the computer. You want to grant an equal cache for each application.
  You simply need to set the “Percentage of the available memory used for the cache” to 33% for each server.

  Default cache size for each application: (768 - 256) x 33% = 170 MB.

- Creation of an executable application intended for wide distribution using multiple configurations. After testing, it is found that the application operates at its best with a cache of between 20 and 100 MB.
  - Physical memory to reserve = 256 MB.
  - Percentage of the available memory used for the cache = 50% so as to leave memory available for any other applications.
  - Minimum size = 20 MB and Maximum size = 100 MB.

  On a machine with 256 MB of RAM, 4D uses the minimum cache = 20 MB.
  On a machine with 1 GB of RAM, 4D uses the maximum cache = 100 MB.
  On a machine with 384 MB of RAM, 4D uses (384 - 256) x 50% = 64 MB.

- **Keep the cache in physical memory for 4D Server and 4D Desktop (on Macintosh)**
  This advanced option can be used to "force" the cache to be kept in the physical memory (RAM) of the machine when the database is executed by 4D Desktop (all versions) or 4D Server under MacOS.

  Under MacOS, by default the cache can be flushed in part or entirely in the virtual memory of the machine when necessary (the virtual memory is an area set aside on the hard disk). Since this mechanism can, in certain cases, noticeably alter the performance of applications deployed with 4D Server or 4D Desktop, you can check this option in order for the cache to be kept permanently in physical memory. Of course, the machine configuration (size of RAM, number of applications running simultaneously, etc.) must be compatible with this principle.

Note This option does not affect the operation of applications run with the 4D Developer application.

- **Flush Data Buffers every … Minutes**: Specifies the time period between each automatic saving of the data cache, i.e., its writing to disk.
  4D saves the data placed in the cache at regular intervals. You can specify any time interval between 1 and 500 minutes. By default, 4D saves your data every 15 minutes. The application also saves your data to disk each time you change to another environment or exit the application.

When you anticipate heavy data entry, consider setting a short time interval between saves. In case of a power failure, you will only lose the data entered since the previous save (if the database is running without a log file).
If there is a noticeable slowing down of the database each time the cache is flushed, you need to adjust the frequency. This slowness means that a huge amount of records is being saved. A shorter period between saves would therefore be more efficient since each save would involve fewer records and hence be faster.

- **Display flush window**: When this option is checked, the 4D application displays a window in the lower left corner of the screen when flushing the data buffers. This operation momentarily blocks user actions and gives them a visual indication that the operation is in progress:

Window indicating that the data cache is being saved

WEDD

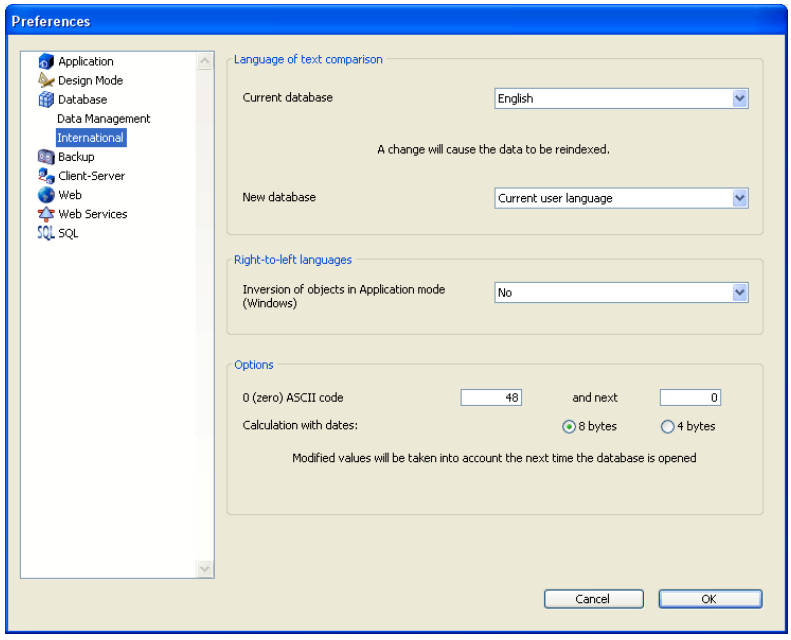
The WEDD area lets you set a link signature between the structure file and the data file for the open database.

A link signature associates a structure file with a data file. A structure file containing a link signature can only be opened with a data file containing the same link signature and vice versa. This way, you can prevent the use of incompatible data files. You can also define link signatures if you work with several databases and you want to avoid accidentally opening a data file that does not belong to a particular structure.

By default, this parameter is not set in databases. To add a link signature to a database, simply enter a custom value in the link signature area. It can be composed of any type of character combination. The structure file and the current data file are then linked.
To remove a link signature, simply delete the contents of this area.

International Page

This page lets you modify various parameters related to the use of the application in a multi-lingual context.



Language of text comparison

This group of parameters can be used to configure the language used for character string processing and comparison.
It is possible to choose a comparison language for the open database (limited requirement) and a default language for new databases (permanent requirement). The language choice has a direct influence on the sorting and searching of text, as well as the character case, but it has no effect on the translation of texts or on the date, time or currency formats, which remain in the system language. By default, 4D uses the system language.

A4D database can thus operate in a language different from that of the system. When a database is opened, the 4D engine detects the language used by the data file and provides it to the language (interpreter or compiled mode). Text comparisons, regardless of whether they are carried out by the database engine or the language, are done in the same language.

After selecting a language in the **Current database** menu and validating the Preferences dialog box, the language code is stored in the structure and data, then the indexes are immediately rebuilt. It is necessary to restart the database in order for comparisons made by the language to take this configuration into account.

When creating a new data file, 4D uses the language previously set in the **New database** menu. When opening a data file that is not in the same language as the structure, the data file language is used and the language code is copied into the structure.

Right-to-left Languages

The term "right-to-left language" refers to those languages that are read from right to left, like Arabic for example.
4D includes extended support for right-to-left languages under Windows. When you activate the "right-to-left" mode, the 4D database interface (forms and menus) are automatically inverted in Application mode.

Note This is valid only under Windows. There is no equivalent function under MacOS X.

This inversion concerns not only the direction text objects are read from within, but also graphic objects, the position of objects in forms, window titles and buttons, the location of menus, and so on.

Note In current versions of 4D, the inverted mode is not available for the printing of forms.

The **Inversion of objects in Application mode** option is used to activate and configure the "right-to-left" mode at the general database level. There are three options that can be used to configure the activation of this mode according to the context in which the database is executed:

- **No**: When this option is selected, the database never switches to "right-to-left" mode, even if the operating system is configured in this mode. This option is selected by default for databases.
- **Automatic**: When this option is selected, the database automatically switches to "right-to-left" mode when the operating system is configured in this mode (this is the case, for example, of the Arabic version of Windows).

- **Yes**: When this option is selected, the database immediately switches to "right-to-left" mode, even if the operating system is configured in the conventional (left-to-right) mode.

  When you modify this option, it is immediately applied to the database (after validation of the Preferences). Keep in mind that this preference is specific to the database.

Notes

• This preference can also be handled by programming using the SET DATABASE PARAMETER command.
• When the right-to-left mode is activated, it is possible to disable it locally for each form (see the "Do Not Invert Objects (Windows)" paragraph on page 403).
• By default, the right-to-left mode does not reverse the contents of pictures in forms. An additional option can be used to invert the pictures when necessary (see the "Mirror Effect for Pictures (Windows)" paragraph on page 460).

Options

The Options area is mainly intended for the configuring of non-Roman-based languages (Japanese, Arabic, etc.).

- **0 (zero) ASCII code**: ASCII code of the character representing zero. On an American or French system, its value is 48. It may be different on another system (Japanese, for example).
- **...and next**: Value of the next byte after zero. This parameter is only used if you are localizing for Arabic or Hebrew systems. In all other cases, set this parameter to 0 (zero).
- **Calculation with dates**: This option is used to adjust the 4D settings for date calculations. In the case of Farsi calendars, the calculation uses 4 bytes. In all other cases, 8 bytes are needed.

Backup Theme

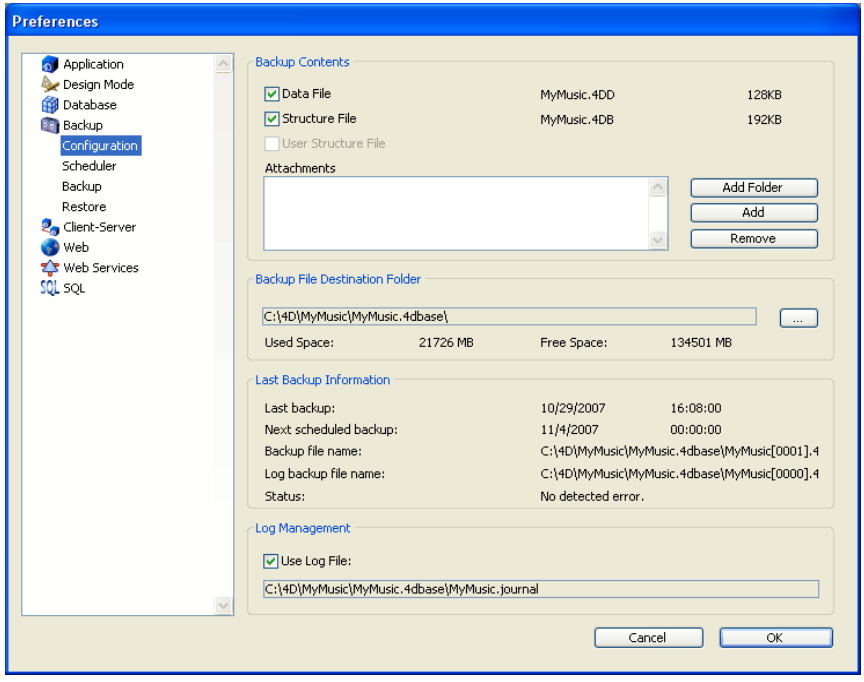
The pages of this theme are used to configure the backup system of the database. For a detailed description of the 4D backup mechanisms, refer to the chapter "Backup and Restoring of the Database" on page 1169.

4D Server

Backup parameters can only be set from the 4D Server machine.

Configuration Page

This page lets you set the backup files and their location as well as that of the log file. It also provides information about the last backup.



Backup Contents

This area allows you to set which files and/or folders to copy during the next backup.

- **Data file**: Database data file. When this option is checked, the current log file of the database, if it exists, is backed up at the same time as the data.
- **Structure file**: Database structure file. In cases where databases are compiled and merged with 4D Desktop, this option allows you to backup the .exe file on Windows and the package on Mac OS.
- **User structure file** (optional): Database User structure file that contains customized user forms (where applicable). For more information, please refer to the chapter “User Forms” on page 685
- **Attachments**: This area allows you to specify a set of files and/or folders to be backed up at the same time as the database. These files can be of any type (documents or plug-in templates, labels, reports, pictures, etc.).

You can set either individual files or folders whose contents will be fully backed up. Each attached element is listed with its full access path in the "Attachments" area.

- **Add Folder**: When you click this button, 4D displays a dialog box that allows selecting a folder to add to the backup. In the case of a restore, the folder will be recovered with its internal structure. You can select any folder or volume connected to the machine, with the exception of the folder containing the database files.
- **Add**: When you click this button, 4D displays a dialog box that allows you to select a file to add to the backup.
  You cannot select a database file as an attached file.
- **Remove**: This button allows you to remove the selected file from the list of attached files.

Backup File Destination Folder

This area lets you view and change the location where backup files as well as log backup files (where applicable) will be stored.

To view the location of the files, click in the area in order to display their pathname as a pop-up menu.

To modify the location where these files are stored, click the [...] button. A selection dialog box appears, which allows you to select a folder or disk where the backups will be placed. The "Used Space" and "Free Space" areas are updated automatically and indicate the remaining space on the disk of the selected folder.

Last Backup Information

The "Last Backup Information" area provides information about the last database backup. Information is provided if at least one backup has taken place.

- **Last backup**: Date and time of the last backup.
- **Next scheduled backup**: Date and time of the next backup; information is provided in this area if a backup schedule has been put into place.
- **Backup file name**: Access path and file name of the last main backup. If the backup is segmented, the name of the first segment is displayed.
- **Log backup file name**: Access path and file name of the last log file backup (if applicable).

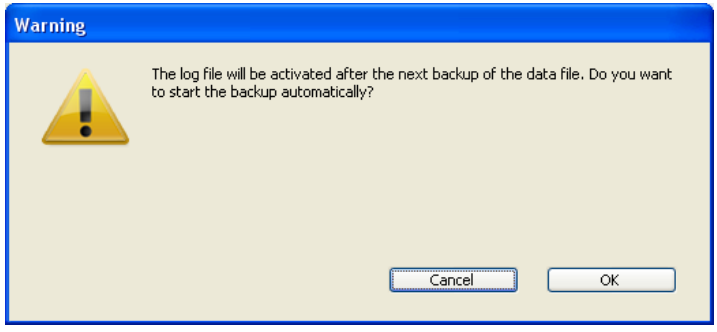
- **Status**: This area displays the error code of the last backup as well as a description of this code. If the backup was executed properly, the area indicates “No detected error.” For scheduled backups, you can use this area to verify that the last backup occurred as scheduled.

Log Management

The **Use Log File** option, when checked, indicates that the database uses a log file. Its pathname is specified below the option. When this option is checked, it is not possible to open the database without a log file.

By default, any database created with 4D uses a log file: the creation of this file is included in the automatic backup strategy (see the “Automatic Backup Strategy” paragraph on page 38). The log file is named ***DataFileName*.journal** and is placed in the folder containing the database structure.

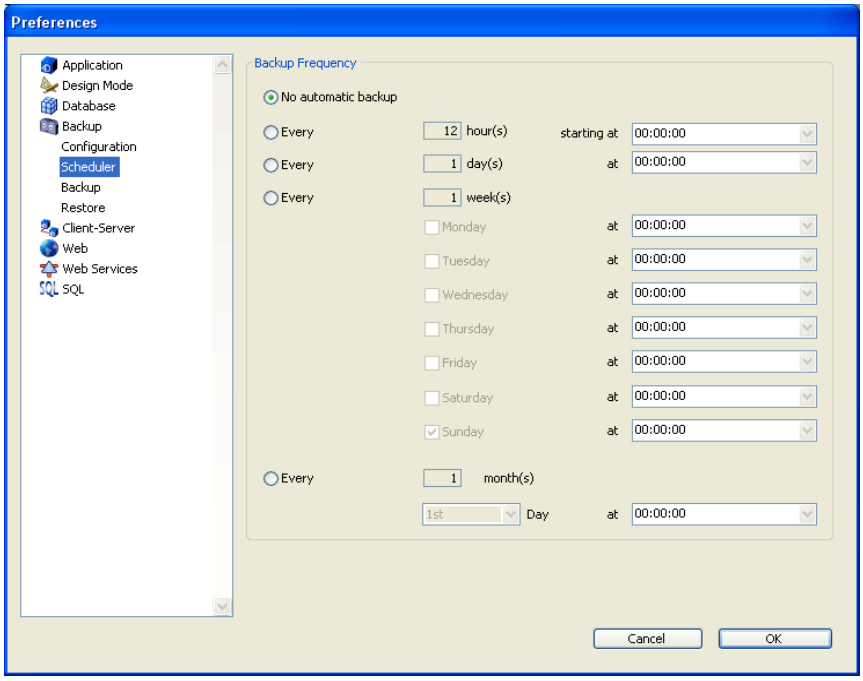
Activating a new log file requires the data of the database to be backed up beforehand. When you check this option and then validate the Preferences window, a warning dialog box informs you that a backup is necessary:



If you click **OK**, the backup begins immediately then the log file is activated. If you click **Cancel**, the request is saved but the creation of the log file is postponed and it will actually be created only after the next backup of the database.

Scheduler Page

This page lets you set and configure scheduled automatic backups of the database.

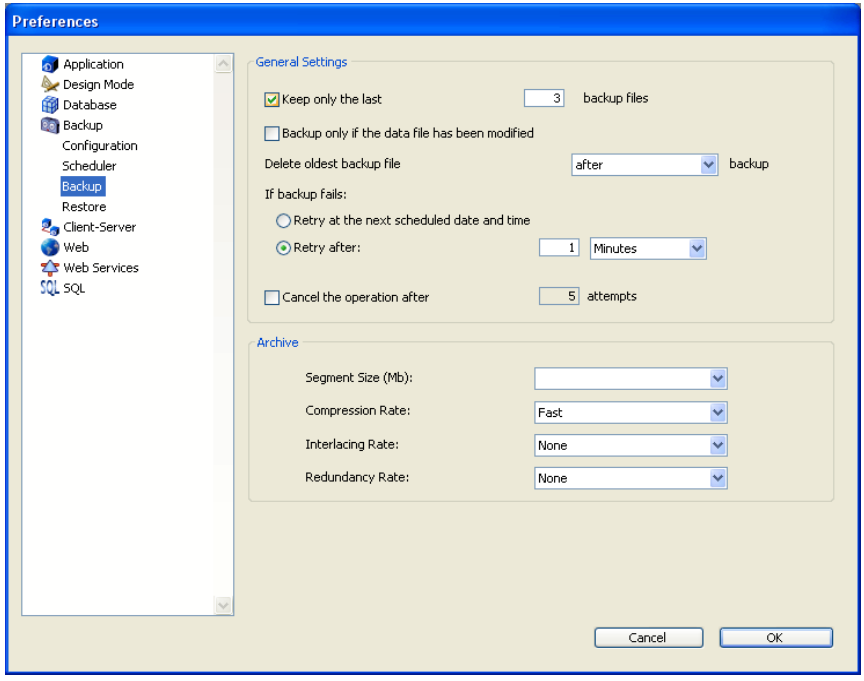


- **No automatic backup**: The scheduled backup feature is disabled.
- **Every X hour(s)**: This option allows programming backups on an hourly basis. You can enter a value between 1 and 24.
  - **starting at**: Sets the time at which the first hourly backup will begin.
- **Every X day(s) at x**: This option allows programming backups on a daily basis. Enter 1 if you want to perform a daily backup. When this option is checked, you must enter the time when the backup should start.
- **Every X week(s) day at x**: This option allows programming backups on a weekly basis. Enter 1 if you want to perform a weekly backup. When this option is checked, you must enter the day(s) of the week and the time when the backup should start. You can select several days of the week, if desired. For example, you can use this option to set two weekly backups: one on Wednesday and one on Friday.

- **Every X month(s), Xth Day at x**: This option allows programming backups on a monthly basis. Enter 1 if you want to perform a monthly backup. When this option is checked, you must indicate the day of the month and the time when the backup should start.

Backup Page

This page lets you set advanced options concerning database backups.



General Settings

This area sets various mechanisms that come into play during backups.

- **Keep only the last *X* backup files**: This parameter allows activating and configuring the mechanism used to delete the oldest backup files, which avoids the risk of saturating the disk drive.
  This feature works as follows: Once the current backup is complete, 4D deletes the oldest archive if it is found in the same location as the archive being backed up and has the same name (you can request that the oldest archive be deleted before the backup in order to save space). If, for example, the number of sets is set to 3, the first three backups create the archives MyBase-0001, MyBase-0002, and MyBase-0003 respectively. During the fourth backup, the archive MyBase-0004 is created and MyBase-0001 is deleted. By default, the mechanism for deleting sets is enabled and 4D keeps 3 backup sets.

To disable the mechanism, simply deselect the option.

Note This parameter concerns both the database backup sets and the log file backup sets.

- **Backup only if the data file has been modified**: When this option is checked, 4D starts scheduled backups only if data has been added, changed or deleted in the database since the last backup. Otherwise, the scheduled backup is cancelled and put off until the next scheduled backup. No error is generated; however the backup journal notes that the backup has been postponed.
  This option also allows saving machine time for the backup of databases principally used for viewing purposes. Please note that enabling this option does not take any modifications made to the structure file or attached files into account.

Notes • When this option is checked and the backup starts manually, 4D displays a dialog box indicating that the data file has not been modified and allows the user to confirm or cancel the operation.
• This parameter concerns both database and log file backups.

- **Delete oldest backup file before/after backup**: This option is only used if the "Keep only the last X backup files" option is checked. It allows you to specify whether 4D should start by deleting the oldest archive before starting the backup (**before** option) or if the deletion should occur once the backup is completed (**after** option). In order for this mechanism to work, the oldest archive must not have been renamed or moved.
- **If backup fails**: This option allows setting the mechanism used to handle failed backups (backup impossible).

Note 4D considers a backup as failed if the database was not launched at the time when the scheduled automatic backup was set to be carried out.

When a backup cannot be performed, 4D allows making a new attempt. Two options are possible:

- **Retry at the next scheduled date and time**: This option only makes sense when working with scheduled automatic backups. It amounts to cancelling the failed backup. An error is generated.

- **Retry after X second(s), minute(s) or hour(s)**: When this option is checked, a new backup attempt is executed after the wait period. This mechanism allows anticipating certain circumstances that may block the backup. You can set a wait period in seconds, minutes or hours using the corresponding menu. If the new attempt also fails, an error is generated and the failure is noted in the status area of the last backup and in the backup journal file.
- **Cancel the operation after X attempts**: This parameter is used to set the maximum number of failed backup attempts.
  If the backup has not be carried out successfully after the maximum number of attempts set has been reached, it is cancelled and the error 1401 is generated ("The maximum number of backup attempts has been reached; automatic backup is temporarily disabled"). In this case, no new automatic backup will be attempted as long as the application has not been restarted, or a manual backup has been carried out successfully.
  This parameter is useful in order to avoid a case where an extended problem (requiring human intervention) that prevented a backup from being carried out would have led to the application repeatedly attempting the backup to the detriment of its overall performance.
  By default, this parameter is not checked.

Archive

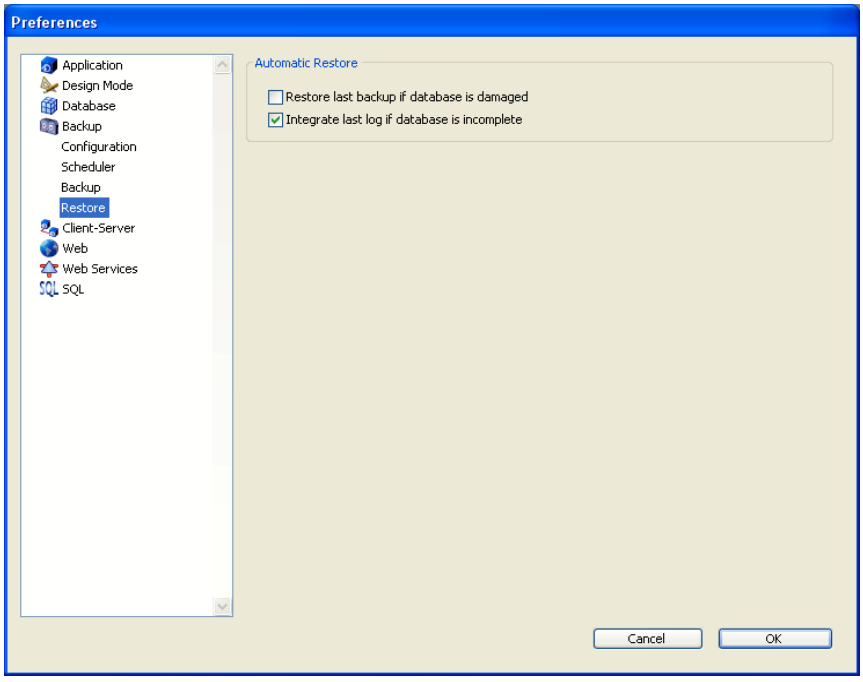
This area allows setting archive generation options. These options apply to main backup files and to log backup files.

- **Segment Size (Mb)**
  4D allows you to segment archives, i.e., to cut it up into smaller sizes. This behavior allows, for example, the storing of a backup on several different disks (CDs, ZIPs, etc.). During restore, 4D will automatically merge the segments. Each segment is called *MyDatabase[xxxx-yyyy].4BK*, where *xxxx* is the backup number and *yyyy* is the segment number. For example, the three segments of the MyDatabase database backup are called MyDatabase[0006-0001].4BK, MyDatabase[0006-0002].4BK and MyDatabase[0006-0003].4BK.
  The **Segment Size** menu is a combo box that allows you to set the size in MB for each segment of the backup. You can choose one of the preset sizes or enter a specific size between 0 and 2048. If you pass 0, no segmentation occurs (this is the equivalent of passing **None**).

- **Compression Rate**
  By default, 4D compresses backups to help save disk space. However, the file compression phase can noticeably slow down backups when dealing with large volumes of data.
  The **Compression Rate** option allows you to adjust file compression:
  - **None**: No file compression is applied. The backup is faster but the archive files are considerably larger.
  - **Fast** (default): This option is a compromise between backup speed and archive size.
  - **Compact**: The maximum compression rate is applied to archives. The archive files take up the least amount of space possible on the disk, but the backup is noticeable slowed.
- **Interlacing Rate** and **Redundancy Rate**
  4D generates archives using specific algorithms that are based on optimization (interlacing) and security (redundancy) mechanisms. You can set these mechanisms according to your needs. For these two options, 4D provides a default rate **Medium**. The menus for these options also contain rates of **Low**, **High** and **None**.
  - **Interlacing Rate**: Interlacing consists of storing data in non-adjacent sectors in order to limit risks in the case of sector damage. The higher the rate, the higher the security; however, data processing will use more memory.
  - **Redundancy Rate**: Redundancy allows securing data present in a file by repeating the same information several times. The higher the redundancy rate, the better the file security; however, storage will be slower and the file size will increase accordingly.

Restore Page

This page is used to configure the automatic mechanisms to be put into play when restoring a database backup.



- **Restore last backup if database is damaged**: When this option is checked, the program automatically starts the restore of the data file of the last valid backup of the database, if an anomaly is detected (corrupted file, for example) during database launch. No intervention is required on the part of the user; however, the operation is logged in the backup journal.

Note In the case of an automatic restore, only the data file is restored. If you wish to get the attached files or the structure file, you must perform a manual restore.

- **Integrate last log file if database is incomplete**: When this option is checked, the program automatically integrates the log file when opening or restoring the database in the following cases:
  - When opening a database, the current log file is automatically integrated if 4D detects that there are operations stored in the log file that are not present in the data. This situation arises, for example, if a power outage occurs when there are operations in the data cache that have not yet been written to the disk.

- When restoring a database, if the current log file or a log backup file having the same number as the backup file is stored in the same folder, 4D examines its contents. If it contains operations not found in the data file, the program automatically integrates it.

The user does not see any dialog box; the operation is completely automatic. The goal is to make use as easy as possible. The operation is logged in the backup journal.

Settings Preferences Using Backup.XML File

All 4D backup and restore preferences are available as an separate XML file. 4D uses the data in this file to display options in the Preferences dialog box and at the moment each backup starts.

It is possible to get and change backup settings using the 4D XML commands or using any XML editor. This allows developers to manage backup settings programmatically, especially with applications compiled and merged with 4D Unlimited Desktop.

The XML backup and restore settings file is named **Backup.XML**. It is automatically created at the following location:
DatabaseFolder/Preferences/Backup/Backup.XML

... where **DatabaseFolder** is the folder containing the database structure file. The subfolders **Preferences/Backup/** are created automatically if necessary.

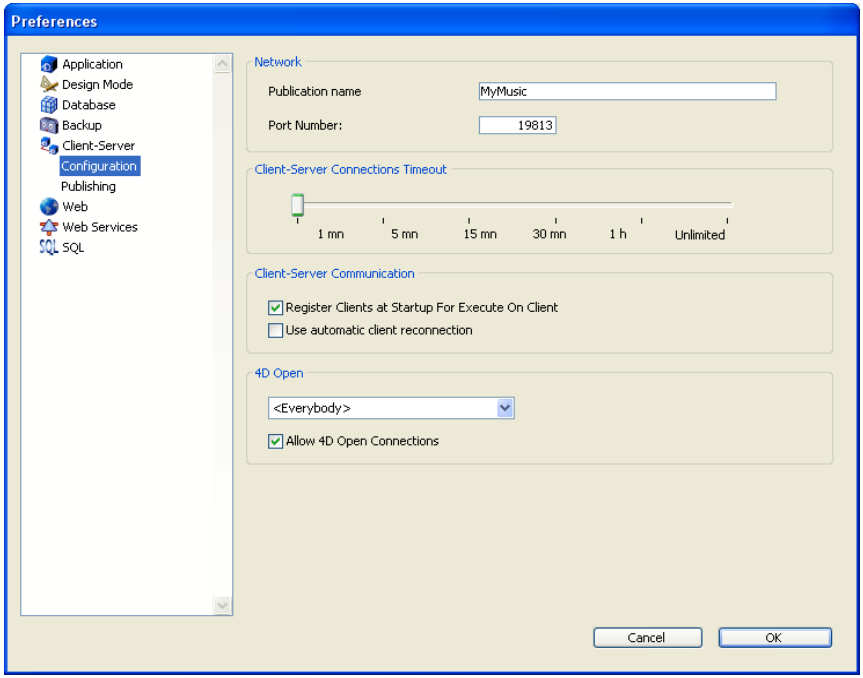
The backup preferences XML keys are described in a separate manual named *XML Keys-Backup*.

Client-Server Theme

The pages of this theme group together parameters related to the use of the database in Client-Server mode. Naturally, these preferences are only taken into account during client-server operation of the database.

Configuration Page

This page is used to set options related to connections with 4D Server.

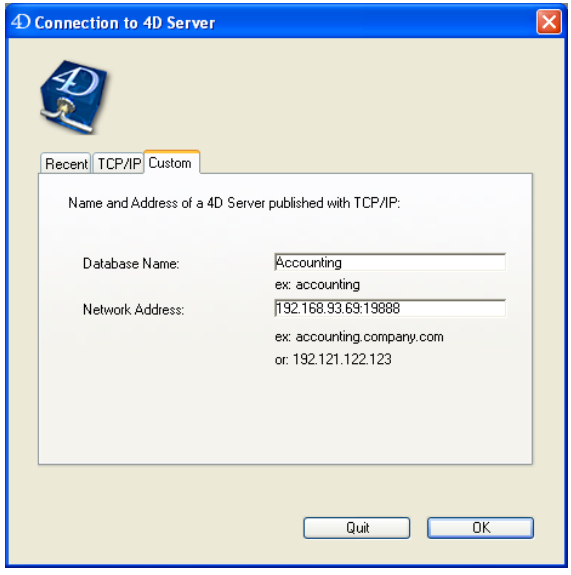


Network

- **Publication name**: This option is used to modify the publication name of the database published by 4D Server, i.e. the name displayed on the dynamic **TCP/IP** publication page of the connection dialog box.
By default, 4D Server uses the database structure file name. You can enter any custom name you want.

Note This parameter is not taken into account for custom client-server applications. In principle, the client application connects directly to the server application, without passing through the connection dialog box. However, in the event of an error, this dialog box appears; in this case, the publication name of the server application is the name of the compiled database.

- **Port Number**: This option allows modifying the TCP port number on which 4D Server publishes the database. This information is stored in the database structure file and on each client machine. By default, the TCP port number used by 4D Server and 4D Client is 19813.
Customizing this value is necessary when you wish to use several 4D applications on the same machine with the TCP protocol; in this case, you must specify a different port number for each application.
When you modify this value from 4D Server or 4D Client, it is automatically carried over to all 4D Client machines connected to the database. To update other client machines that are not connected, during their next connection, simply enter the new port number (preceded by a colon) after the IP address of the server machine on the **Custom** page of the connection dialog box. For example, if the new port number is 19888:



Note Only databases published on the same port as the one set in 4D Client are visible in the **TCP/IP** dynamic publishing page.

Client-Server Connections Timeout

This thermometer is used to set the *timeout* (period of inactivity beyond which the connection is closed) between 4D Server and the client machines connecting to it.
The **Unlimited** option removes the *timeout*. When this option is selected, client activity control is eliminated and an inactive client will remain connected indefinitely.

When a timeout is selected, the server will close the connection of a client if it does not receive any requests from the latter during the specified time limit.

Client-Server Communication

This area lets you set options relating to communications between the client machines and the server.

- **Register Clients at Startup For Execute On Client**: This option lets 4D Server automatically register each 4D Client machine as soon as it connects to the 4D Server database. Once a client is registered, it can perform any processing that was requested by the server or another client.
  This option is mainly designed for the use of the **Execute on Client** function when run from the Execute Method dialog box (for more information about this, refer to the "From the Execute Method Dialog Box" paragraph on page 828).
  If you want to implement a sophisticated task allocation system, you should consider using the language commands that were designed for this purpose (for more information, refer to the *4D Language Reference* manual).
  You can set this option in the Preferences of the client or server machine. In both cases it will apply to each client that connects to the database since it is stored in the structure file of the database. If this option is modified, you have to quit and reconnect all the clients that were already connected in order for the modification to be taken into account.

- **Use automatic client reconnection**: This low-level function allows, in some specific configurations, the automatic reconnection of client machines in the case of unexpected disconnection.
  When this option (global for all client machines) is checked, the reconnection feature is automatically activated if a client connection with the server is lost. There is only one attempt: if the reconnection fails, an error -10002 is returned. If it is successful, an alert dialog box is displayed and a notification icon appears in the Windows task bar.

Note For the automatic client reconnection feature to be available, the server-side timeout must be set to a value higher than 1 minute.

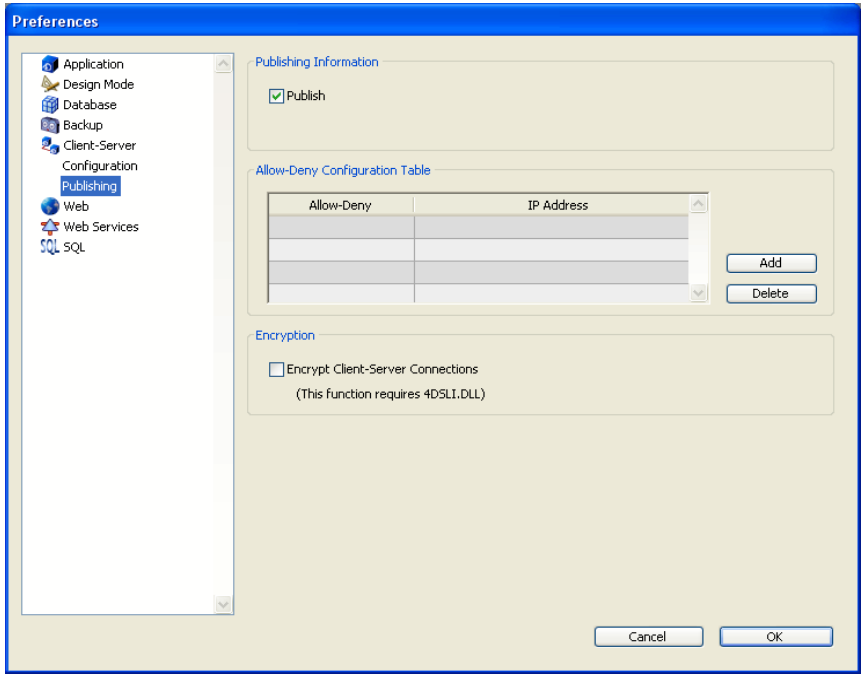
4D Open

4D Open is the API (Application Programming Interface) that allows non-4D Client applications to connect to 4D Server.

- **4D Open Access**: This menu lets you specify the group of users allowed to connect to the 4D Server database via 4D Open, when the "Allow 4D Open Connections" option is checked. For more information about creating groups of users, refer to the chapter "Users and Groups" on page 831.
- **Allow 4D Open Connections**: When it is checked, this option gives the group specified by the "4D Open Access" menu the possibility of connecting to a 4D Server from a 4D Open application.

Publishing Page

This page includes several options related to the publication of the database by 4D Server.



Publishing Information

- **Publish**: This option lets you indicate whether or not the 4D Server database will appear in the list of published databases.
  - When this option is checked, the database is made public and appears in the list of published databases (default option),

- When the option is not checked, the database is not made public and it does not appear in the list of published databases (**TCP/IP** page). To connect, users must manually enter the address of the database on the **Custom** page of the connection dialog box.

Allow-Deny Configuration Table

This table allows you to set access control rules for the database depending on 4D Client machine IP addresses. This option allows reinforcing security, for example, for strategic applications.

Note This configuration table does not control Web connections.

The behavior of the configuration table is as follows:

- The "Allow-Deny" column allows selecting the type of rule to apply (**Allow** or **Deny**) using a pop-up menu. To add a rule, click on the **Add** button. A new row appears in the table. The **Delete** button lets you remove the current row.
- The "IP Address" column allows setting the IP address(es) concerned by the rule. To specify an address, click in the column and enter the address in the following form: 123.45.67.89.
  You can use an * (asterisk) character to specify "starts with" type addresses. For example, 192.168.* indicates all addresses starting with 192.168.
- The application of rules is based on the display order of the table. If two rules are contradictory, priority is given to the rule located highest in the table.
  You can re-order rows by modifying the current sort (click the header of the column to alternate the direction of the sort). You can also move rows using drag and drop.
- For security reasons, only addresses that actually match a rule will be allowed to connect. In other words, if the table only contains one or more Deny rules, *all* addresses will be refused because none will match at least one rule. If you want to deny only certain addresses (and allow others), add an **Allow *** rule at the end of the table.
  For example:
  • Deny 192.168.* (*deny all addresses beginning with 192.168*)
  • Allow * (*but allow all other addresses*).

By default, no connection restrictions are applied by 4D Server: the first row of the table contains the **Allow** label and the * (all addresses) character.

Encryption

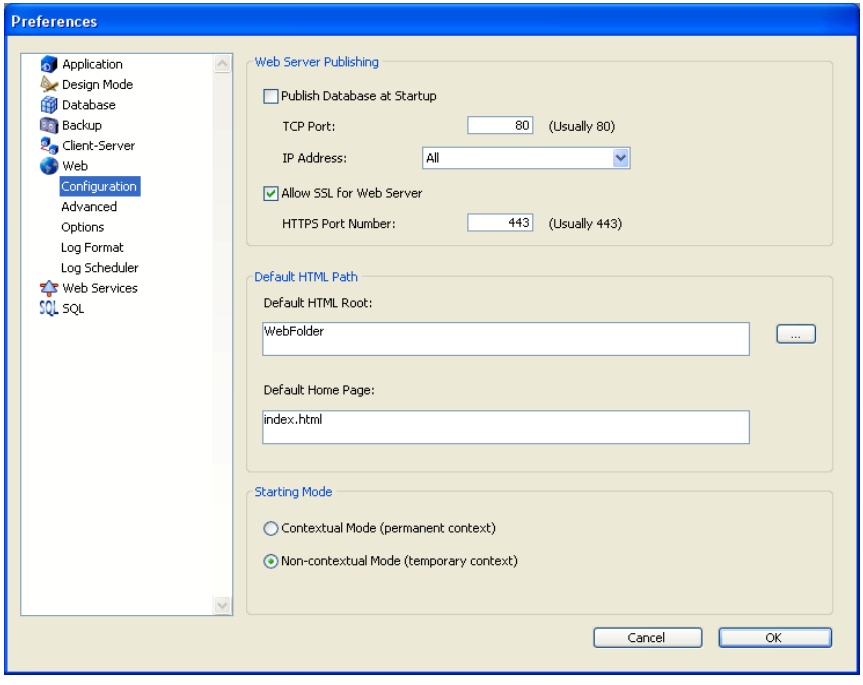
- **Encrypt Client-Server Connections**: This allows you to activate or disable 4D Server connection encryptions. In fact, the “classic” client-server architecture can benefit from the encryption features proposed by the SSL protocol. Its use allows you to reinforce communication security but it will slow down connections. This option does not require any additional settings.
  By default, this option is not selected.
  For more information, please refer to the *4D Language Reference* manual.

Web Theme

The pages of this theme are used to configure the operation of the integrated Web server of 4D (security, startup, connections, etc.). For more information about the Web server, refer to the “Web Server” chapter of the *4D Language Reference* manual.

Configuration Page

This page groups together the initial configuration settings of the Web server.



Web Server Publishing

- **Publish Database at Startup**: This option controls the automatic default publication of the database on the Web, i.e., the launching of the Web server on application startup. If this option is checked, 4D accepts Web connections. If this option is not checked, the database is not published by default. However, you can publish a database at any time using the **Run** menu or using the language.

- **TCP Port**: Sets the TCP port to be used when the Web server is started. The default value is 80. Setting the TCP port allows you to run several Web servers on the same computer. To do so, select different TCP ports for each Web server. This option also allows you to let the OS provide Web services on port 80, while you are using 4D as a Web server on another port.

Note When the TCP port is different from the default port, you must add the port number to the IP address in order to be able to access the Web server. The syntax to use is *IPaddress:TCPport*. For example,123.45.67.89:8080 indicates that the TCP port used is 8080.

- **IP Address**: This option allows you to specify the IP address on which the Web server will receive the HTTP requests. By default, the Web server answers on all the IP addresses (**All** option).
  The pop-up menu automatically displays all the IP addresses present on the machine. If you want the server to only respond to requests sent to a specific TCP/IP address, simply select this address from the menu.
  This feature is for 4D Web Servers located on machines with multiple TCP/IP addresses, which is frequently the case for most Internet host providers.

- **Allow SSL for Web Server**: This allows you to activate or deactivate SSL protocol usage for Web server connections. By default, this option is selected.
  You can deselect this option if you choose not to use SSL functions in your Web server or if another Web server authorizing secured connections is active on the same machine. For more information, please refer to the *4D Language Reference* manual.

- **HTTPS Port Number**: Sets the TCP/IP port used by the Web server for secured HTTP connections over SSL (HTTPS protocol). By default, the HTTPS port number is set to 443 (standard value).
  You may consider changing this port number for two main reasons:
  - For security reasons: Hacker attacks against Web servers are generally concentrated on standard TCP ports (80 and 443).
  - Under Mac OS X, to allow "standard" users to launch the Web server in a secured mode. Under Mac OS X, the use of TCP/IP ports reserved for Web publications (0 to 1023) requires specific access privileges: only the root user can launch an application using these ports. In order for standard users to launch the Web server, one solution is to modify the TCP/IP port number.

  You can pass any valid value. To learn the standard TCP port numbers, please refer to the section *Appendix B, TCP port numbers* in the 4D Internet Commands documentation.

Default HTML Path

- **Default HTML Root**: This option allows you to set the folder in which 4D will search for the static HTML pages and the pictures to send to the browsers. Moreover, the HTML root folder sets the hierarchical level above which the files will not be accessible.
  By default, 4D sets a default HTML root folder named "WebFolder." This setting automatically activates the access restriction system. If it does not already exist, the folder is created on disk when the Web server is launched for the first time.

  If you keep the default location, the root folder is created:
  - With 4D Developer and 4D Server, at the same level as the database structure file.
  - With 4D Client, at the same level as the 4D Client .exe file (under Windows) or the software package (under Mac OS).

  To change the location of the Default HTML Root folder, enter the pathname relative to the folder that you want to set as the new location. You must use HTML syntax (Unix). For more information, refer to the "Connection Security" section of the *4D Language Reference* manual.

- **Default Home Page**: This option lets you set a default home page for all browsers that connect to the database. If you do not modify this parameter, 4D creates a standard page named index.html, which is used more particularly to test the Web server.
For more information, refer to the "Web Server Settings" section of the *4D Language Reference* manual.

Starting Mode

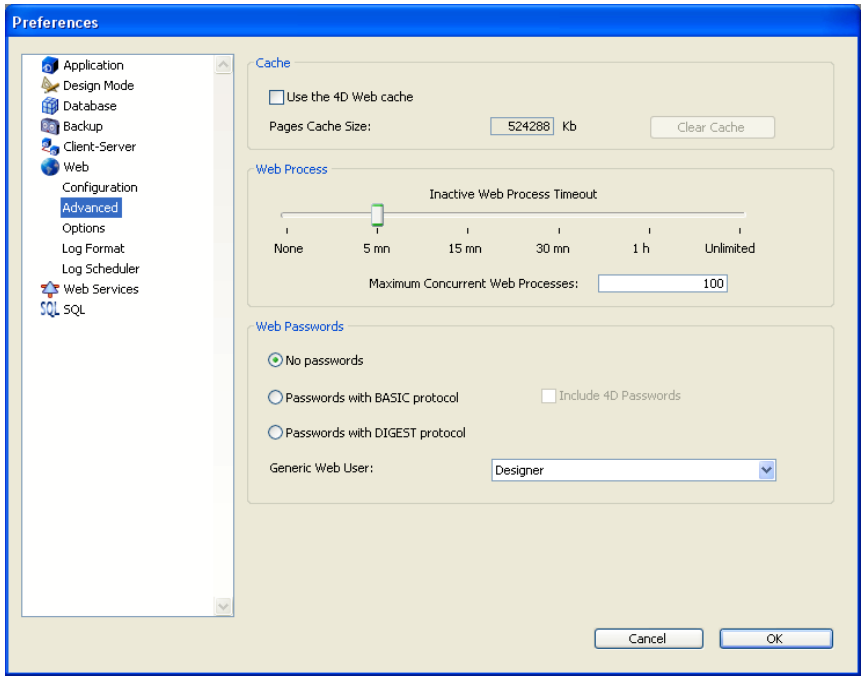
This option lets you set the mode, **contextual** or **non-contextual**, in which the Web server must start. By default, the Web server starts in non-contextual mode.
For more information, refer to the "Using the Contextual Mode" section of the *4D Language Reference* manual.

Note This option is not available with 4D Client because the contextual mode is not supported by the 4D Client Web server.

- **Reuse Temporary Contexts**: This option (found only in the Preferences of 4D Client) is used to optimize Web server operation in 4D Client by recycling Web processes created for handling previous Web requests. In fact, the Web server of 4D Client needs a specific Web process for the handling of each Web request; when necessary, this process connects to the 4D Server machine in order to access the data and database engine. It then generates a temporary context using its own variables, selections, etc. Once the request has been dealt with, this process is killed.
When the **Reuse Temporary Contexts** option is checked, 4D maintains the specific Web processes created on 4D Client and reuses them for subsequent requests. By removing the process creation stage, Web server performance is improved. In return, you must make sure in this case to systematically initialize the variables used in 4D methods in order to avoid getting incorrect results. Similarly, it is necessary to erase any current selections or records set during the previous request.

Advanced Page

This page is used to set advanced configuration parameters for the 4D Web server.



Cache

This area allows you to set the use of the Web server cache.

- **Use the 4D Web cache**: This option lets you activate the 4D Web server cache. This cache (not to be confused with the 4D data cache) lets you load static pages, GIF pictures, JPEG pictures (<100 KB) and style sheets into memory. Using the cache allows you to significantly increase the Web server's performance when sending static pages. The cache is shared by all the Web processes.
  By default, this cache is not activated.
- **Pages Cache Size**: This option lets you set the size of the 4D Web server cache. This value will depend on the number and size of the static pages of your Web site, as well as the resources of the host machine. Once the cache has been enabled, the 4D Web server first looks in the cache for the page requested by the browser. If it finds the page, it sends it immediately. If not, 4D loads the page from disk and places it in the cache. When the cache is full and additional space is required, 4D "unloads" the oldest pages first, among the least requested ones.

- **Clear Cache**: Clicking this button clears the cache of the pages and pictures that it contains (if, for example, you have modified a static page and you want to reload it in the cache).

Web Process

- **Inactive Web Process Timeout**: Lets you set the timeout (period of inactivity beyond which the connection will be closed) for Web connection processes (contextual mode only).
  The **Unlimited** option eliminates activity control of the browsers. An inactive client will remain connected indefinitely. However, when a timeout is set, the Web server will close the connection if it does not receive any requests from the client within the specified time interval. For more information about this, refer to the "Using the Contextual Mode" section of the *4D Language Reference* manual.
- **Maximum Concurrent Web Processes**: Defines the strict upper limit of the number of Web processes created by the Web server. This setting includes all types of Web processes: contextual, non-contextual or those belonging to the pool[1] of processes. This setting allows you to avoid 4D Web server saturation that can happen when there is a massive number of requests or an excessive demand for creating contexts. By default, this number is set to 32,000 (in other words, up to 31,999 Web processes can be simultaneously created). You can set any value between 10 and 32,000. In theory, the maximum number of processes is the result of dividing Available memory by Web process pile size.

  Another solution consists in consulting the information about Web processes displayed in the Runtime Explorer: the current number of Web processes and the maximum number reached since the Web server launch are indicated (see the section "Watch Page" on page 137).
  When the maximum number of concurrent Web processes is reached, 4D will no longer create any new processes and returns HTTP status 503 - Service Unavailable to each new request.

Note The maximum number of Web processes can also be set using the SET DATABASE PARAMETER command.

1. The "pool" of Web processes allows you to increase Web server reactivity in non-contextual mode. This reserve is resized with a minimum (0 by default) and a maximum (10 by default) of processes to recycle. These values can be modified using the **SET DATABASE PARAMETER** command (selectors 6 and 7). Once the maximum number of pool Web processes has been set, if it is lower than the upper limit of the "pool," this limit will be lowered to the maximum number of Web processes.

Web Passwords

This group of parameters is used to specify the identification mode of connections to the 4D Web server. For more information, please refer to the "Connection Security" section of the 4D *Language Reference* manual.

- **No passwords**: No authentication is carried out for connections to the Web server.
- **Passwords with BASIC protocol**: Standard authentication in BASIC mode (passwords entered by the users are not encrypted in HTTP requests). When this option is selected, for each connection, a dialog box for entering the user name and password is displayed by the browser. These two values, as well as the connection parameters (IP address and port, URL, etc.) are sent to the On Web Authentication database method so that you can process them.
- **Include 4D Passwords**: This option is enabled only when the previous one has been checked. It allows you to use, instead of or in addition to your own password system, the password system of the database (as defined in 4D).

Notes With the 4D Client Web server, every site published by 4D Client machines will share the same table of users. In fact, the validation of users/passwords is carried out by the 4D Server application.

- **Passwords with DIGEST protocol**: Authentication in Digest mode. If you check this option, you will need to manage the connections using the new Validate Digest Web Password command in the On Web Authentication database method.
  Unlike Basic mode, the Digest mode is not compatible with standard 4D passwords: it is not possible to use 4D passwords as Web IDs. The "Include 4D passwords" option is dimmed when this mode is selected. The IDs for Web users must be managed in a customized manner (for example, via a table).

Note You must restart the Web server for the changes made to these parameters to be taken into account.

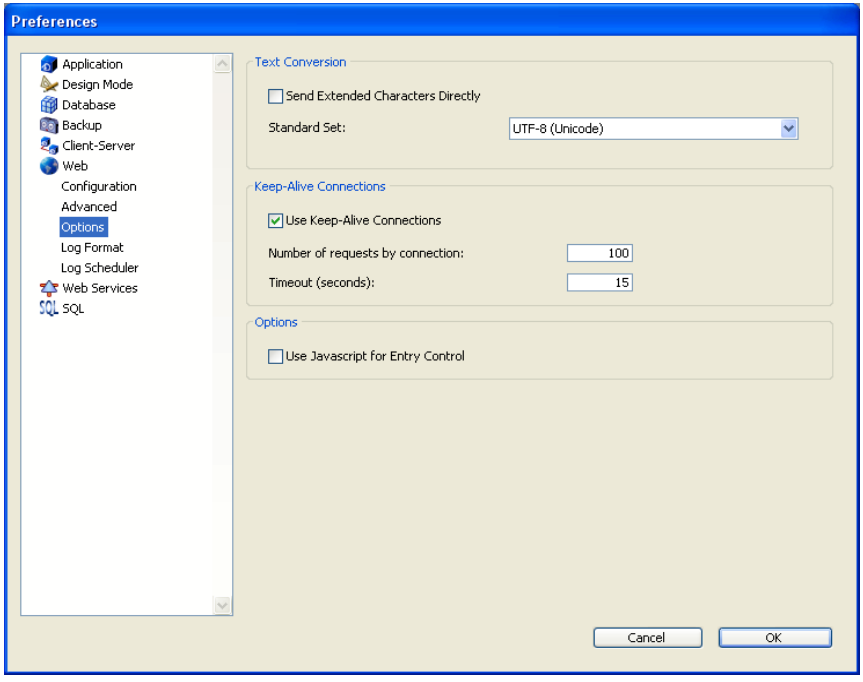
- **Generic Web User**: When a user has been set in this menu, each browser that connects to the database will benefit from the access rights and restrictions associated with this user. This provides you with a simple way of controlling the browser's access to different parts of the database. By default, the Generic Web User is the Designer and browsers have full access to the entire database.

Note Do not confuse this option, which allows you to restrict the browser's access to different parts of the database (tables, menus, etc.), with the Web server's connection control system, managed by the password system and the On Web Authentication database method (see above).

When the "Include 4D Passwords" option is selected and the user who connects is defined in 4D's password access system, the "Generic Web User" option is ignored and the users connect under their own access privileges.

Options Page

This page is used to set text conversion and keep-alive connection options.



Text Conversion

- **Send Extended Characters Directly**: By default, the 4D Web server converts extended characters found on the Web pages to HTML entities before sending them. They are then interpreted by the browsers. When it is checked, this option lets you send extended characters "as is," without conversion into HTML entities.
  This option can save you a considerable amount of time on foreign systems (in particular, Japanese ones).

- **Standard Set**: This option allows you to choose the character set 4D uses to communicate with Web browsers connecting to the database. The value selected in the drop-down list will determine the conversion of characters that the 4D Web server performs when sending or receiving an HTML document (dynamic or static pages). The default character set for the US version of 4D is ISO-8859-1, which corresponds to the standard Western (Latin1) encoding.

  4D lets the Web browser know which encoding is to be used for HTML information. It is therefore not necessary to set this particular setting in each browser.

Note The set of characters defined in this menu is also used for HTML exporting of Quick reports (see the "Printing to an HTML Document" paragraph on page 753) and for the Mac to ISO language command.

Keep-Alive Connections

This area is used to configure the TCP keep-alive connections for the 4D Web server.

When the Web server uses keep-alive connections, a single TCP connection is kept open for the set of exchanges between the Web browser and the server in order to save system resources and to optimize transfers. For more information about this mechanism, refer to the "Web Server Settings" section of the *4D Language Reference* manual.

- **Use Keep-Alive Connections**: This option is checked by default. In most cases, it is recommended that you keep this option checked because it accelerates transfers.
  If the Web browser does not support keep-alive connections, the 4D Web server automatically switches to HTTP/1.0.

The **Number of requests by connection** and **Timeout** options allow you to set the operation of the keep-alive connections, when they are enabled:

- **Number of requests by connection**: Allows you to set the maximum number of requests and responses able to travel over a keep-alive connection. Limiting the number of requests per connection allows you to prevent server flooding due to a large number of incoming requests (a technique used by hackers).
  The default value (100) can be increased or decreased depending on the resources of the machine hosting the 4D Web server.
- **Timeout**: This value sets the maximum wait period (in seconds) during which the Web server maintains an open TCP connection without receiving any requests from the Web browser. Once this period is over, the server closes the connection.
  If the Web browser sends a request after the connection is closed, a new TCP connection is created automatically. This operation is transparent to the user.

Options

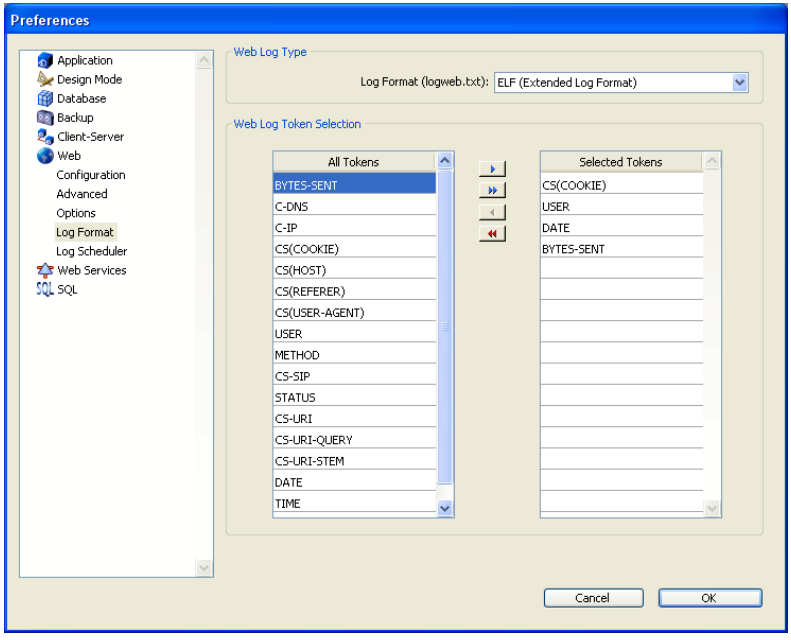
- **Use Javascript for Entry Control**: When this option is checked, part of the browser's entry control is taken over by automatic Java scripts. The data entry controls and the data types (fields or variables) to which they can be applied are as follows:
  - Minimum value (for numeric values);
  - Maximum value (for numeric values);
  - Mandatory value (for numeric and alphanumeric values).

  Generated Java scripts, which are small in size, display alert dialog boxes during data entry without preventing the user from validating the entry (which is still 4D's responsibility):

Log Format Page

This page can be used to activate and set the Web log file format (*logweb.txt*). The *logweb.txt* file allows after-the-fact analysis of requests sent to the 4D Web server. Once it has been activated, it is automatically placed:

- With 4D Developer and 4D Server, next to the structure file,
- With 4D Client or an executable application, next to the application (Windows) or package (Mac OS).



Note The Web log file can also be activated and deactivated via programming, using the SET DATABASE PARAMETER command.

- **No Log File**: When this option is selected, 4D will not generate a Web log.
- **CLF (Common Log Format)**: When this option is selected, the Web log will be generated in CLF format. The CLF format cannot be customized.
- **DLF (Combined Log Format)**: When this option is selected, the Web log will be generated in DLF format.
  DLF format is similar to CLF format and uses exactly the same structure. It simply adds two additional HTTP fields at the end of each request: *referer* and *user-agent*.

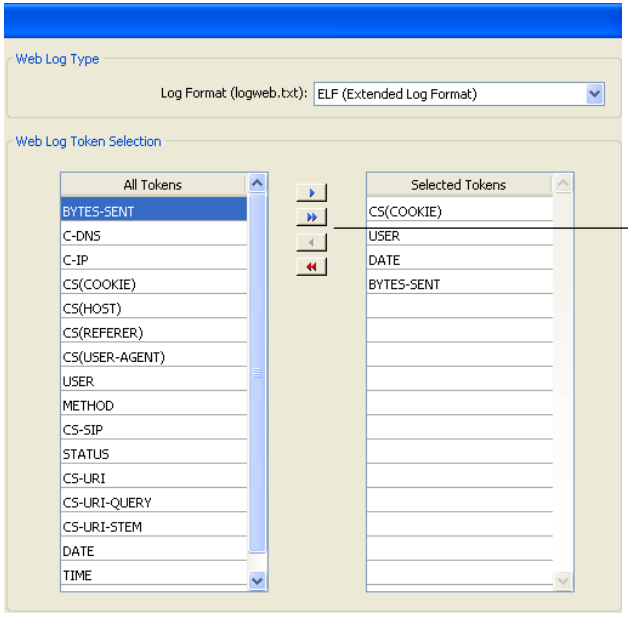
- *referer*: Contains the URL of the page pointing to the requested document.
- *user-agent* : Contains the name and version of the browser or software of the user at the origin of the request.

The DLF format cannot be customized.

- **ELF (Extended Log Format)**: When this option is selected, the Web log will be generated in ELF format.
The ELF format is very widespread in the world of HTTP browsers. It can be used to build sophisticated logs that meet specific needs. For this reason, the ELF format can be customized: it is possible to choose the fields to be recorded as well as their order of insertion into the file.
- **WLF (WebStar Log Format)**: When this option is selected, the Web log will be generated in WLF format.
WLF format was developed specifically for the WebSTAR server. It is similar to the ELF format, with only a few additional fields. Like the ELF format, it can be customized.

Configuring Fields

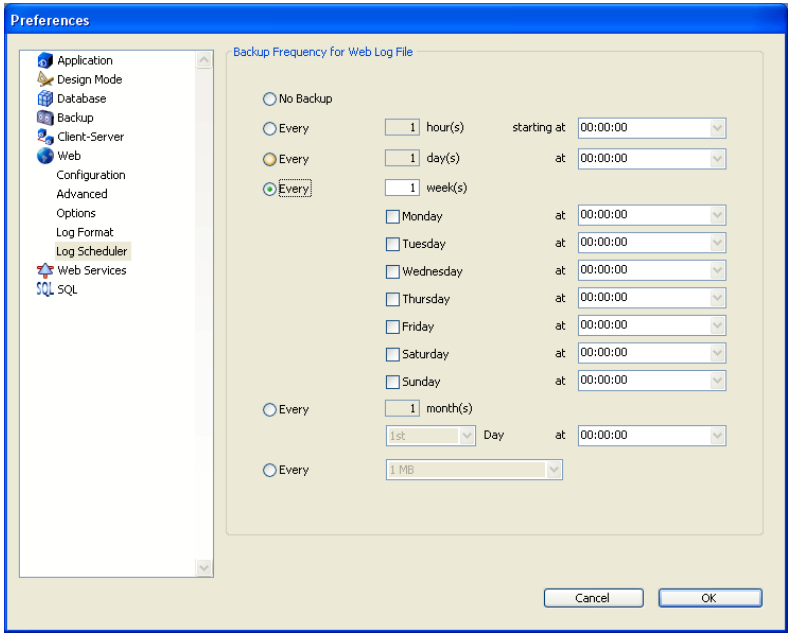
When you choose the ELF (Extended Log Format) or WLF (WebStar Log Format) format, the "Web Log Token Selection" area displays the fields available for the chosen format. You will need to select each field to be included in the log. To do so, use the arrow buttons or simply drag and drop to move the desired fields:



For more information about the fields of the various formats, please refer to the “Information about the Web site” section in the 4D *Language Reference* manual.

Log Scheduler Page

This page is used to set the automatic backup parameters for the Web log:



First you must choose the frequency (days, weeks, etc.) or the file size limit criterion by clicking on the corresponding radio button. You must then specify the precise moment of the backup when required.

- **No Backup**: The scheduled backup function is deactivated.
- **Every *X* hour(s)**: This option is used to program backups on an hourly basis. You can enter a value between 1 and 24.
  - **starting at**: Used to set the time at which the first backup will begin.
- **Every *X* day(s) at *X***: This option is used to program backups on a daily basis. Enter 1 if you want to perform a daily backup. When this option is checked, you must indicate the time when the backup should start.

- **Every *X* week(s), *day* at *X***: This option is used to program backups on a weekly basis. Enter 1 if you want to perform a weekly backup. When this option is checked, you must indicate the day(s) of the week and the time when each backup should start. You can select several days of the week if desired. For example, you can use this option to set two weekly backups: one on Wednesdays and one on Fridays.
- **Every *X* month(s), *X*th day at *X***: This option is used to program backups on a monthly basis. Enter 1 if you want to perform a monthly backup. When this option is checked, you must indicate the day of the month and the time when the backup should start.
- **Every *X* MB**: This option is used to program backups based on the size of the current Web log file. A backup is automatically triggered when the file reaches the set size. You can set a size limit of 1, 10, 100 or 1000 MB.

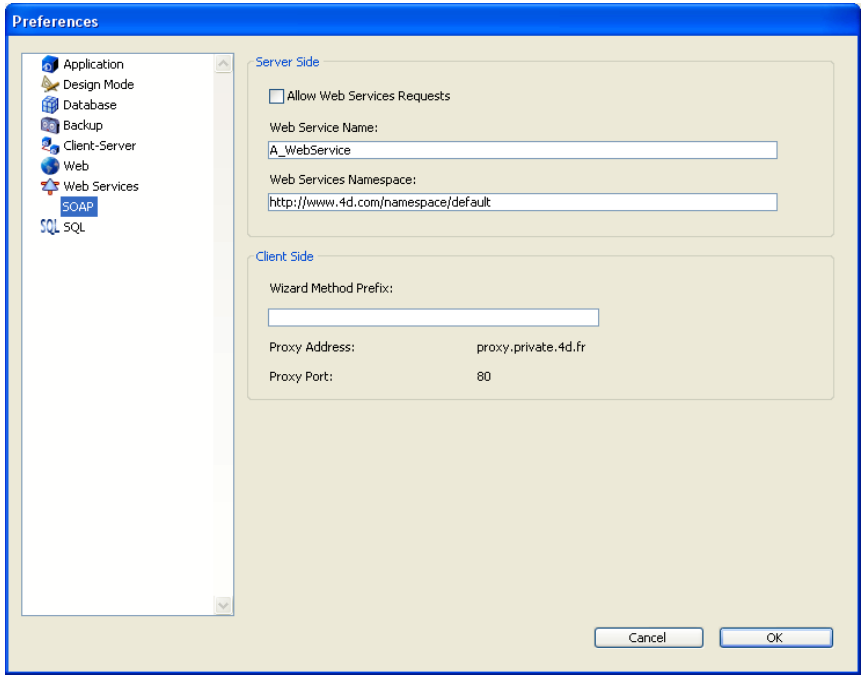
Note In the case of scheduled backups, if the Web server was not launched when the backup was scheduled to occur, on the next startup 4D considers the backup as failed and applies the appropriate settings, set in the database Preferences.

Web Services Theme

This theme contains a single page, used to configure the publication and use of Web Services in your 4D database.
For more information about Web Services, refer to the chapter "Publishing or Using Web Services" on page 1137.

SOAP Page

The SOAP page groups together all the configuration options for the Web Services, both server and client side.



Server Side

This area contains various options related to the use of 4D as a Web Services "server," i.e., publishing project methods in the form of Web Services.

- **Allow Web Services Requests**: This option lets you "initialize" the publication of Web Services. If this option has not been checked, 4D refuses SOAP requests and does not generate a WSDL — even if the methods have the **Offered as a Web Service** attribute. When this option is checked, 4D creates the WSDL file.
- **Web Service Name**: This area lets you change the "generic name" of the Web Service. This name is used to differentiate the services both at the SOAP server level (when the server publishes several different Web Services), as well as in the Web Services directories. By default, 4D uses the name A_WebService.

- **Web Services Namespace**: This area is used to change the namespace of the Web Services published by 4D.
  Each Web Service published on the Internet must be unique. The uniqueness of the names of Web Services is ensured by using XML namespaces. A namespace is an arbitrary character string used to identify a set of XML tags in a unique way. Typically, the namespace begins with the URL of the company (http://mycompany.com/mynamespace). In this case, it is not indispensable to have anything in particular at the URL indicated; what matters is that the character string used is unique. By default, 4D uses the following namespace: http://www.4d.com/namespace/default.

Note In conformity with the XML standard for tag names, the character strings used must not contain spaces nor start with a number.

Client Side

This area contains various options related to the use of 4D as a Web Services "client," i.e., subscribing to services published on the network.

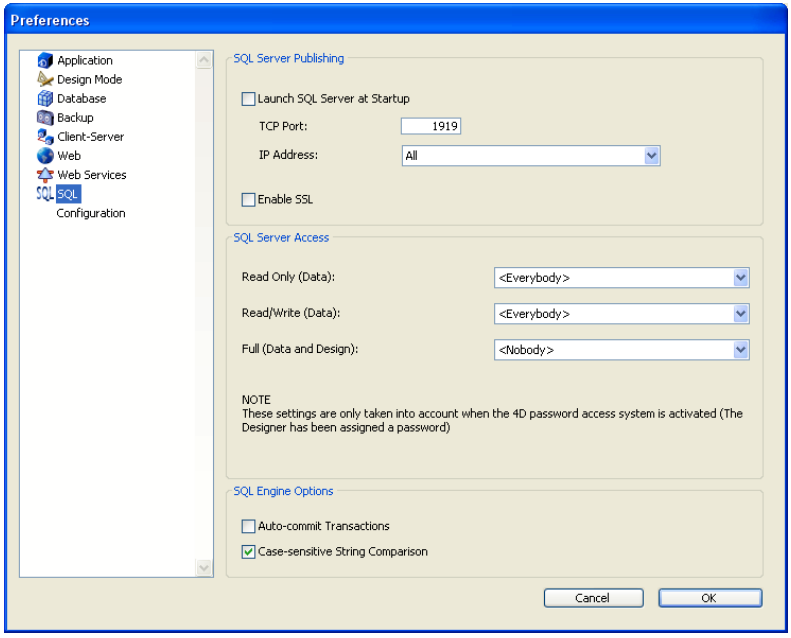
- **Wizard Method Prefix**: This area lets you change the prefix that is added automatically by 4D to the name of proxy methods generated by the Web Services Wizard. Proxy project methods form a link between the 4D application and the Web Services server.
  By default, 4D uses the prefix "proxy_".
- **Proxy Address** and **Proxy Port**: This area lets you check the current Internet connection parameters, used for the subscription to Web Services. These values are only read by 4D; if you want to modify them, you must do so via the Internet parameters of the machine.

SQL Theme

This theme has a single page that is used to configure the functioning of the integrated SQL server of 4D. For more information about the SQL server, please refer to the "Management of 4D SQL Server" paragraph on page 1165.

Configuration Page

This page is used to configure the default publishing parameters, the access rights and the auto-commit mode of the integrated SQL server of 4D:



SQL Server Publishing

- The **Launch SQL Server at Startup** option can be used to start the SQL server on application startup.
- **TCP Port**: By default, the 4D SQL server responds on the TCP port 1919. If this port is already being used by another service or if your connection parameters require another configuration, you can change the TCP port used by the 4D SQL server.

Note If you pass 0, 4D will use the default TCP port number, i.e. 1919.

- **IP Address**: You can set the IP address of the machine on which the SQL server must process SQL queries. By default, the server will respond on all the IP addresses (**All** option).
  The "IP Address" drop-down list automatically contains all the IP addresses present on the machine. When you select a particular address, the server will only respond to queries sent to this address. This is intended for 4D applications hosted on machines having several TCP/IP addresses.

Note On the client side, the IP address and the TCP port of the SQL server to which the application connects must be correctly configured in the ODBC data source definition.

- **Enable SSL**: This option indicates whether the SQL server must enable the SSL protocol for processing SQL connections.

SQL Server Access

You can configure three separate types of access to the 4D database via the SQL server:

- **Read Only (Data)**: Free access to read all the data of the database tables but no adding, modifying or removing of records, nor any modification to the structure of the database is allowed.
- **Read/Write (Data)**: Read and write (add, modify and delete) access to all the data of the database tables but no modification of the database structure is allowed.
- **Full (Data and Design)**: Read and write (add, modify and delete) access to all the data of the database tables as well as modification of the database structure (tables, fields, relations, etc.) is allowed.

  You can designate a set of users for each type of access. There are three options available for this purpose:

  - **<Nobody>**: If you select this option, the type of access concerned will be refused for any queries, regardless of their origin. This parameter can be used even when the 4D password access management system is not activated.
  - **<Everybody>**: If you select this option, the type of access concerned will be allowed for all queries (no limit is applied).

- *Group of users*: This option lets you designate a group of users as exclusively authorized to carry out the type of access concerned. This option requires that 4D passwords be activated. The user at the origin of the queries provides their name and password when connecting to the SQL server via ODBC.

WARNING: *This mechanism is based on 4D passwords. In order for the SQL server access control to come into effect, the 4D password system must be activated (a password must be assigned to the Designer). For more information about this point, please refer to the* *chapter "Users and Groups" on page 831*.

Note An additional security option can be set at the level of each 4D project method. For more information, please refer to the "Available through SQL" paragraph on page 777.

SQL Engine Options

These options are used to set the functioning of the SQL engine in 4D.

- **Auto-commit Transactions**: This option can be used to activate the auto-commit mechanism in the 4D SQL engine. The purpose of the auto-commit mode is to preserve the referential integrity of the data. When this option is checked, any SELECT, INSERT, UPDATE and DELETE (*SIUD*) queries not already carried out within a transaction are automatically included in an ad hoc transaction. This guarantees that the queries will be executed in their entirety or, in the case of an error, completely cancelled.

  Queries already included in a transaction (custom management of referential integrity) are not affected by this option.

  When this option is not checked, no automatic transaction is generated (except for the SELECT... FOR UPDATE queries). By default, this option is not checked.

  You can also manage this option by programming using the SET DATABASE PARAMETER command.

Note Only local databases queried by the 4D SQL engine are affected by this parameter. In the case of external databases, the auto-commit mechanism is handled by the remote SQL engines.

- **Case-sensitive String Comparison**: This option can be used to modify the case sensitivity of characters in SQL queries. It is checked by default, which means that the SQL engine differentiates between upper and lower case letters when comparing strings (sorts and queries). For example “ABC”= “ABC” but “ABC” # “Abc.”

  In certain cases, for example to align the functioning of the SQL engine with that of the 4D engine, you may want string comparisons not to be case sensitive (“ABC”=“Abc”). To do this, you simply need to deselect this option.

  You can also manage this option by programming using the SET DATABASE PARAMETER command.

4 Database Structure

This chapter tells you how to use 4D's Structure editor to create and modify database structures. The structure of a database consists of tables and fields. If a database has more than one table, the structure might include the relations between the tables.

The structure of a database is like the foundation of a house — it provides the basis for everything else. This chapter gives a general description of databases and of how to design database structures to meet different information management needs.

This chapter tells you how to:

- Manipulate table images in the Structure editor,
- Customize the structure display and carry out searches,
- Create tables and set table properties,
- Create fields and set field types and properties,
- Name tables and fields,
- Create and manage indexes,
- Relate tables,
- Import and export structure definitions.

Database Basics

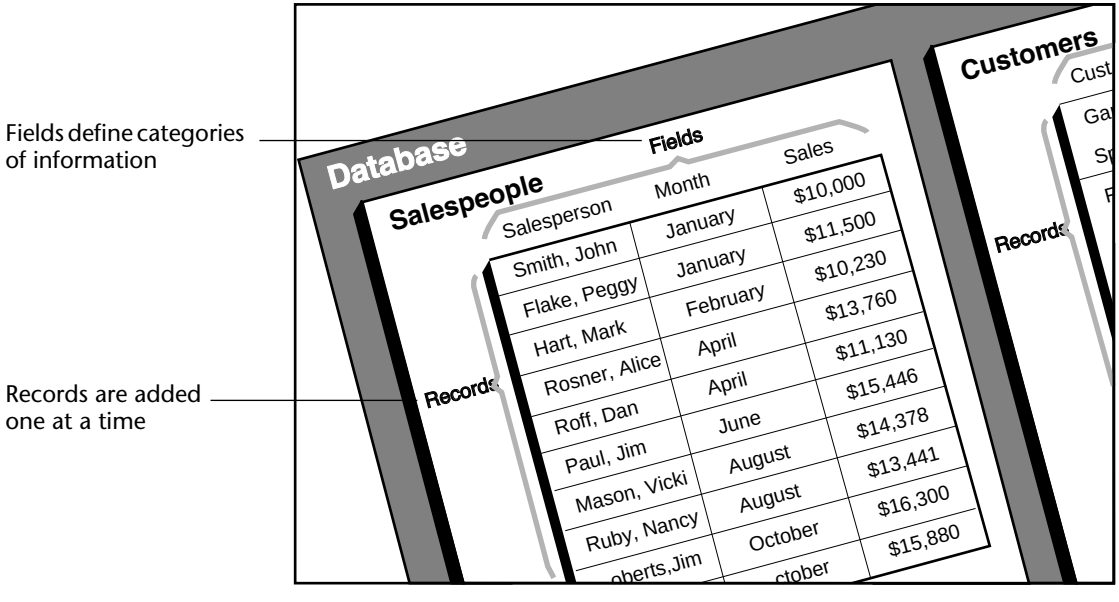
A database is any collection of information organized so that it can be used efficiently. A telephone directory is a good example of a database. So is a dictionary, calendar, or cookbook.

The information in a database is organized in the form of records. Each record contains all of the information about one person or thing in the database. For example, each record in a telephone directory contains one person's name, address, and telephone number.

Each record contains fields. A field is used to store a particular piece of information. For example, in the telephone directory database, one field contains the person's name; a second field contains the person's address and a third field contains the person's telephone number. Every record contains each of these fields and every record can have information in these fields.

A field name usually identifies the information that goes into the field. A field name is usually something like Name, Address, or Phone Number. Each field has a field type that identifies the kind of information that can be entered in a field: numbers, dates, alphanumeric characters, and others. Because each field contains a specific type of data, you can perform calculations and other operations on the information in the fields. For example, numbers from two fields can be added. A date in one field can be compared to a date in another field. A person's first name (stored in one field) can be displayed in front of the last name (stored in another field) to make the first line of an address label.

All the records together make up a *table*. Each database can contain many tables. The following figure shows how these concepts are related.



4D can reorganize records and perform calculations on the information so that the information is useful. For example, 4D can calculate the total values in a field and present the total in a report. It can calculate a total for each salesperson and display a graph that compares sales figures.

Tables

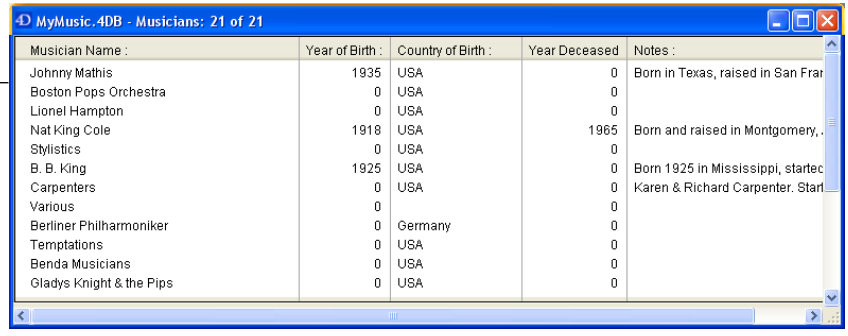
4D can create from 1 to 32 767 tables per database. This means that you can create a structure that is precisely adapted to your needs.

Single-Table Structures

Some databases use only one table. You use a single table for a single category of information such as employees, companies, or inventory. You can have as many fields in a table as you need (up to 32,767).

Table image in the Design environment

Records of the table



In the figure above, every person's record needs the same types of data. The database grows in accordance with the number of employees stored.

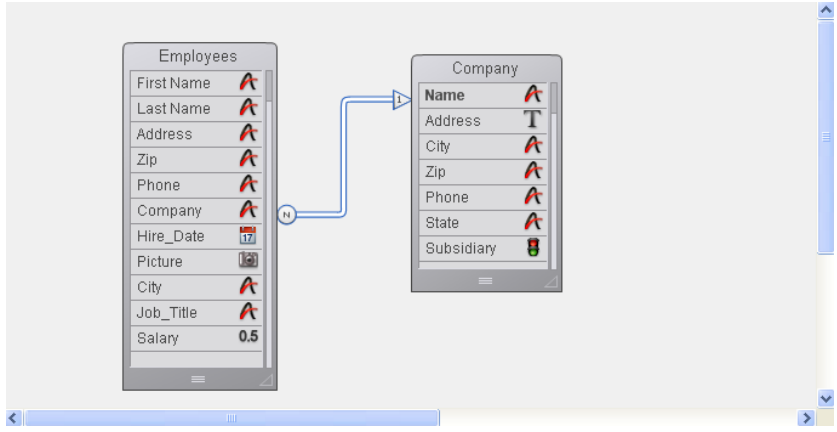
Multiple-Table Structures

A database can often store and access data more efficiently by using more than one table. A good rule to remember is that different types of information should be stored in different tables.

A database that keeps track of both employees and companies is a good example. The records for the employees and the companies are stored in different tables. If the address of a company changes, you need only change that company's record. You do not need to enter that new address for every employee who works for the company.

With a single table, you would have needed to enter the address in each individual record; with two tables, you need to enter that information only once. When a company name is entered in an employee's record, 4D can search for the company's record and automatically display the correct address.

The figure below shows the structure of a multiple-table database in which two tables are related. The arrow drawn between the Company field and the Name field shows that relationship.



The data for each employee is stored in the [Employees] table. Data about each company is stored in the separate [Company] table[1].

4D is called a *relational database* application because it can use multiple tables and relate them in various ways. For example, you can create a report for the [Employees] table that searches the [Company] table and automatically displays and prints information about each employee's company. The relationship between the tables allows information from each table to be available to the report.

You can also enter data directly into related tables. For example, an invoicing database can write information to a [Line Items] table from within an Invoicing screen. You can also write data to related tables using 4D's language.

For information on creating and using related tables, see the section "Relating Tables" on page 295.

1. In the 4D documentation, table names are shown in brackets. This is how they appear in the Method editor.

Sometimes you need a multiple-table structure in which tables are not directly related. It may be convenient to have one database store different kinds of information such as a contact list and an expense table.

4D allows up to 32,767 tables in each database. A table can have up to 32,767 fields. Using multiple tables, virtually any kind of database structure is possible.

Creating a Database Structure

Every database has a structure that consists of at least one table and at least one field. These items must exist before the database can store records.

▶ Typically, you create a structure with the following basic steps:

1 **Create a new database.**
For more information, see the section “Creating a Database” on page 34.

2 **Create a table.**
For more information, see the section “Creating and Modifying a Table” on page 256.

3 **Create fields for the table.**
For more information, see “Creating a Field” on page 265.

4 **Specify the indexes used for each table.**
For more information, see the section “Management of Indexes” on page 288.

5 **Repeat steps 2 to 4 as necessary.**

6 **Relate one table to another, if necessary.**
For more information, see “Relating Tables” on page 295.

The remainder of this chapter describes these steps in detail.

Using the Structure Editor

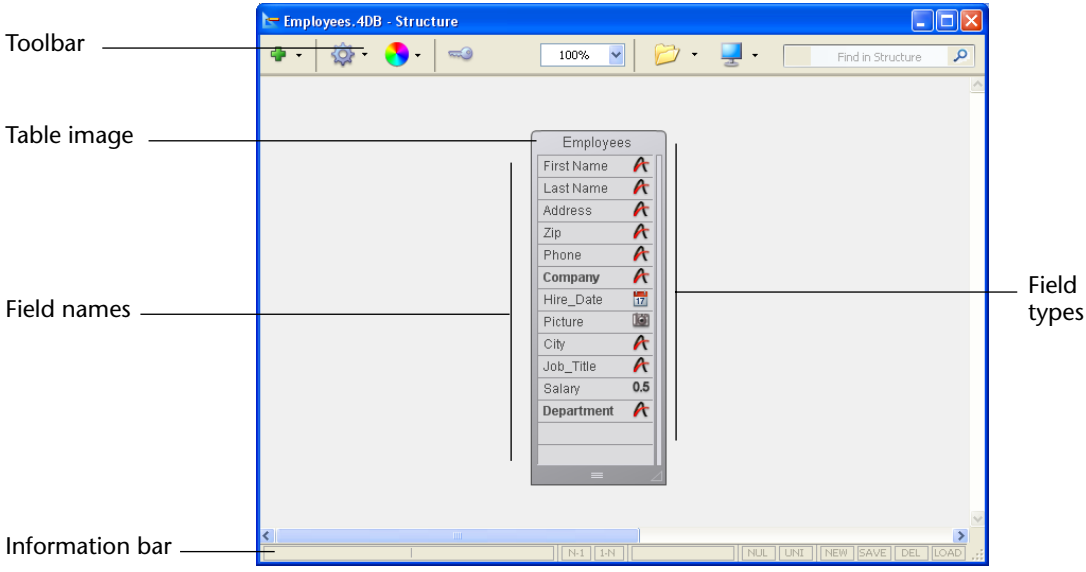
The Structure editor lets you manage the database structure — the tables and their relations. It gives you control over such things as tables, table properties, fields, field properties, and table relations.

A floating Inspector palette can be used to view and modify the properties of structure objects and of the structure editor itself.

The Structure editor gives you a graphic view of a database's structure as well as a toolbar and contextual menus that can be used to carry out database design operations.

Each table is represented by a table image in the Structure editor. It shows the fields and their types, in the form of icons. An information bar displays the characteristics of tables and fields as the mouse moves over them.

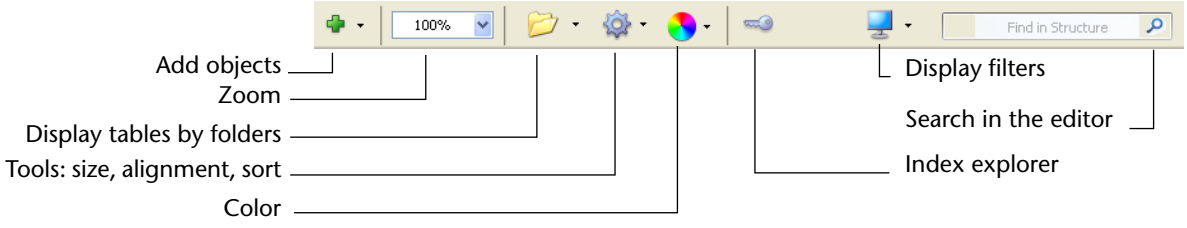
The following illustration shows the Structure editor window with one table image.



Toolbar and Information Bar

The structure editor has a toolbar containing functions like the addition of objects, as well as navigation and display options:

Structure editor toolbar



The functions of the toolbar are described throughout this chapter.

The lower part of the editor window is an information bar displaying data corresponding to the area the mouse is moving over: table, field or relation.

Information about a table

Table name and number

Table triggers:
NEW = On saving new record
SAVE = On saving an existing record
DEL = On deleting a record
LOAD = On loading a record

Information about a field

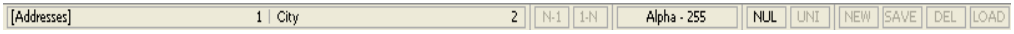


Table name and number

Field name and number

Field type

Field attributes:
NUL = Map NULL values to blank values
UNI = Unique

Information about a relation

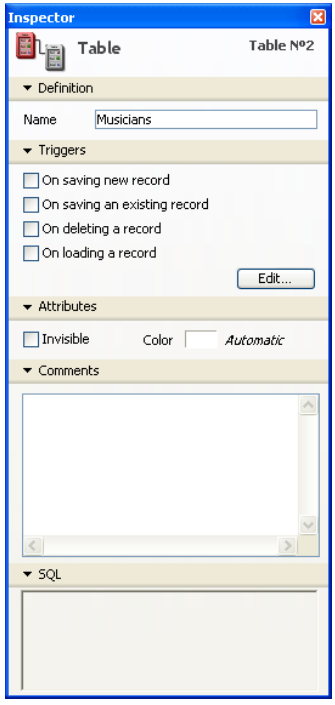
Origin field (Many field)

Destination field (One field)

Relation attributes:
N-1 = Auto Relate One
1-N = Auto One To Many

Inspector Palette

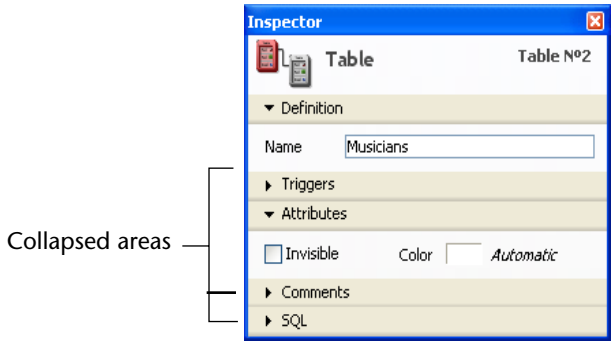
The properties of the Structure editor objects (tables, fields and relations) can be viewed and modified via the Inspector palette:



This palette appears when you double-click on an object. When it is displayed, its contents are updated dynamically depending on the objects selected.

The Inspector displays the general properties of the Structure editor window when you click in an empty area of the window (see the "Customizing the Background of the Editor Window" paragraph on page 254).

The different areas of the palette can be expanded/collapsed by clicking on the adjacent triangles in order to display or hide the corresponding information:



The following shortcuts can be used:

- **Shift+click** on the title bar of a collapsed panel expands that panel and collapses all the others.
- **Alt** (Windows) or **Option** (Mac OS) + **click** on the title bar of a collapsed panel expands all the panels.
- **Alt** (Windows) or **Option** (Mac OS) + **click** on the title bar of an expanded panel collapses all the panels.

The position of the palette and the expanded/collapsed states of its panels are saved.

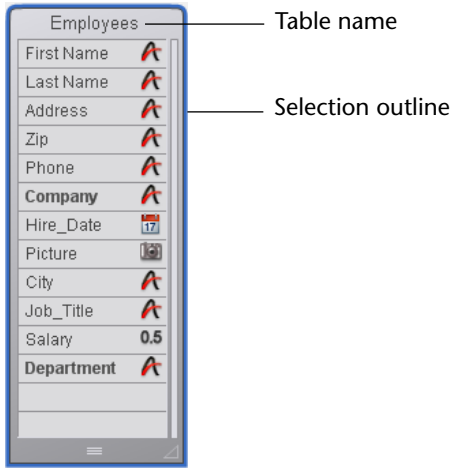
Selecting an Object

To work with the image of an object in the Structure editor, you will first need to select it. You can then specify its properties, move it, resize it, delete it, etc. It is also possible to select several objects of the same type simultaneously in order, for example, to display or modify their common properties in the Inspector palette. You can even select fields from different tables.

► To select a table:

1 **Click the image of the table**
OR
Press [Tab] or Shift+[Tab] to select each table in the Structure editor successively.

The selected table is outlined in blue. Subsequent actions affect the selected table. The figure below shows a selected table.



▶ To select a field or relation:

1 **Click on the field or relation.**
OR
When a table is selected, press the ↑ or ↓ keys to select each field of the table successively.
OR
When a table is selected, press the Home or End key to select the first or last field of the table.

▶ To select several objects of the same type (tables, fields, relations):

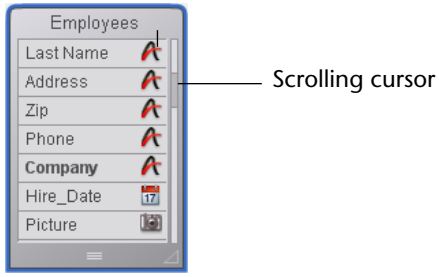
1 **Use Shift+click to select several adjacent objects.**
OR
Use Ctrl+click (Windows) or Command+click (Mac OS) to select several non-adjacent objects.
OR
Click in an empty area and draw a rectangle around the objects to be selected (selection of tables only).
OR
Choose the Select All command in the Edit menu or in the contextual menu of the editor (selection of tables only).

Working with Table Images

You can resize or move the table images in the Structure editor according to your requirements.

Scrolling the Field List

When you add fields to a table or when you reduce the size of a table, there may be more fields in the table than the image can display. When this happens, 4D automatically adds a cursor than you can use to scroll through the list of visible fields:

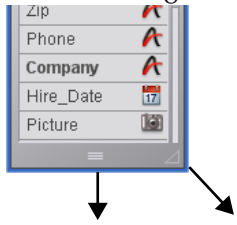


You can also use the mouse wheel (the cursor must be placed above the table).

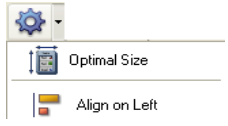
Resizing a Table Image

You can resize the table image in order, for example, to display more field names or to reorganize your screen. There are several possibilities available:

- **Manual resizing**: Simply click and drag the bottom of the table image or its lower right corner.



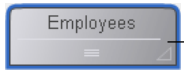
- **Optimal Size**: This automatically resizes the selected table(s) so that their size corresponds exactly to that of the fields they contain (no empty lines are displayed).
  To apply the **Optimal Size** command to one or more tables, you can:
  - Select the **Optimal Size** command in the menu associated with the tool bar button of the editor (this command is not activated if there is no table selected).



  - Select the **Optimal Size** command from the editor's contextual menu (click on a table),

- **Shift**+double-click in the name area of the table to resize. If you repeat this combination, the following sequence is applied: original size -> optimal size - > collapse,

- **Collapse**: This function collapses selected table image(s) so that only their names are visible. This can be useful with large structures.



To apply the **Collapse** command to one or more tables, you can:

- Select the **Collapse** command from the editor's contextual menu (click in the title area of a table).
- **Shift**+double-click in the name area of the table to be resized. If you repeat this shortcut, the following sequence is applied: original size -> optimal size - > collapse.

Notes • A standard double-click on the table name area opens the Inspector.
• **Alt** (Windows) or **Option** (Mac OS) + double-click opens the trigger method of the table in the Method editor.
• **Ctrl** (Windows) or **Command** (Mac OS) + double-click opens the Explorer on the Form page.

4D Server If you resize a table image using 4D Server, the table is resized for all the users in the Design environment.

Moving Table Images

You can move table images in order to regroup them according to function or to reorganize the Structure editor window. You can also align them to improve the readability of the structure. Any relations are redrawn to correspond with the new locations.

► To move a table image manually:

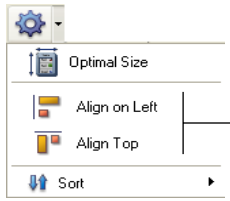
1 **Click on the name of a table and drag it using the mouse cursor.**

Drag the table name bar only. Dragging other parts of the table image may produce different effects, such as creating a new table relation or changing the size of the table image.

4D Server If you move a table image when using 4D Server, the table appears in its new location for all users in the Design environment.

► To align two or more tables:

1 **Select the tables to align and choose an alignment option in the tools menu:**



These commands are only active when at least two tables are selected.

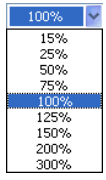
Scrolling

Specific scrolling functions facilitate navigation among large structures:

- The mouse wheel can be used to scroll the contents of the editor window vertically. This will also scroll through the fields of a table when the cursor is placed above it.
- Holding down **Shift** while using the mouse wheel scrolls the contents of the window horizontally.
- Pressing **Shift** activates the “hand” tool which can be used to drag the entire contents of window by clicking in the empty areas.

Zoom

You can modify the display scale of the database structure using the **Zoom** menu in the structure toolbar:



100% is the default value on opening of a database. Zoom action is focused on the selection if there is one.

The current zoom setting is specific to each user. It is memorized when the window is closed.

Note An option in the Preferences can be used to configure the graphic rendering of the Structure editor during a zoom. For more information, please refer to the “Structure Page” paragraph on page 179.

Object Types Display

The **Display** button in the toolbar of the Structure editor is associated with a menu and lets you choose the objects to be displayed in the structure according to their type:

By default, all objects are displayed.

This feature allows various representations or views — from the simplest to the most complete — and provides an analysis tool for the structure based on the level of information required.

The choices are applied to all the tables and fields. They are saved per user and memorized when the window is closed.

Hiding certain objects does not change the position of the tables.

- When several types of fields are checked, a logical OR operator is used to determine the objects to be displayed. For example, if the **Invisible Fields** and **Not Indexed Fields** options are checked, all the non-indexed fields (visible or not) and all the invisible fields (indexed or not) are displayed.
- Tables have priority over fields: if a table is not displayed, its fields are not displayed.
- If you add an object whose type is not displayed (table, field or relation), the object is displayed in the editor, you must select the corresponding option again in the menu associated with the **Display** button in order to hide it.

Highlight/Dim Tables by Folder

You can hide groups of tables in the Structure editor based on folders that are defined in the Explorer window.

Note For more information about folders, please refer to the "Home Page" paragraph on page 103.

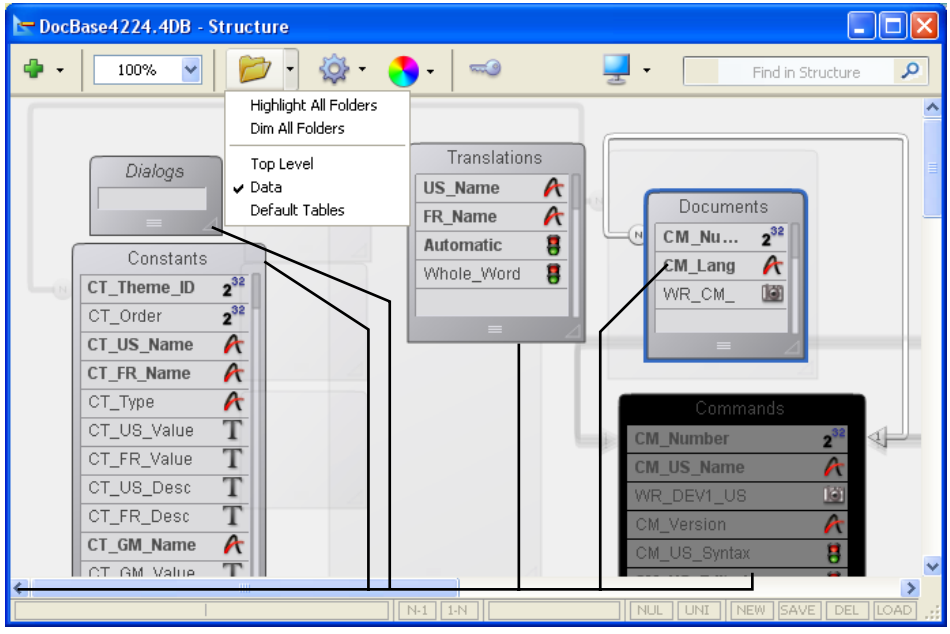
The new **Folders** button in the toolbar can be used to configure the display of the tables.

The menu associated with the button displays the commands that manage the display as well as the list of folders specified in the database. A check will appear next to each folder displayed. You can modify the current display by selecting or deselecting a folder using this menu.

The **Highlight All Folders/Dim All Folders** commands can be used to display/hide all the tables in folders of the database.

Each click on this button  will invert the display of the tables: highlighted tables are dimmed and vice versa.

When the tables are dimmed, by default, only their outlines appear in the Structure window. Only tables that are not dimmed remain entirely visible:



Tables of the Data folder

Appearance of Dimmed Tables

An option lets you set the appearance of dimmed tables (by folder) in the structure editor: **Dimmed** or **Invisible**. This option can be found on the Design Mode/Structure page of the Preferences (see the "Structure Page" paragraph on page 179).

Note It is necessary to close and reopen the Structure editor window in order for this preference to be taken into account.

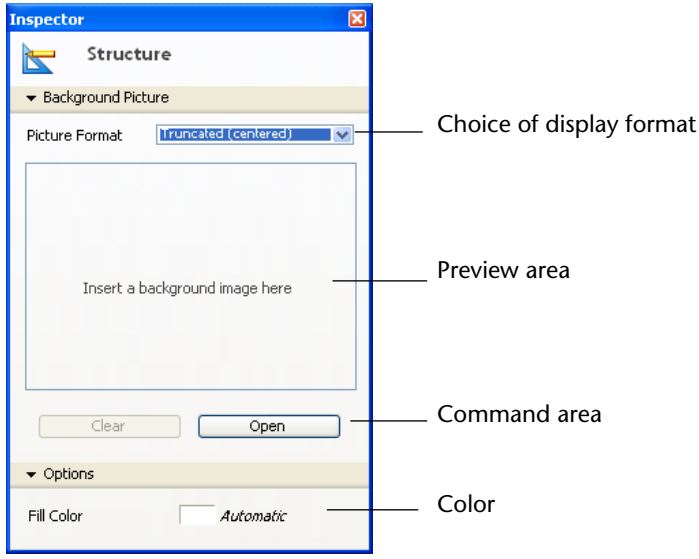
Customizing the Background of the Editor Window

The Structure editor window has specific properties that can be set using the Inspector.

To display the properties of the Structure editor window, you can either:

- Double-click (or click, if the window is already displayed) in an empty area of the window,
- Click with the right mouse button in an empty area of the window and choose the **Structure Properties** command from the contextual menu.

The Inspector palette displays the Structure properties:



The following properties can be set:

- **Background Picture**: You can change the background picture used as well as its display format.

- To change the picture, click on the **Open...** button or click with the **right button** of the mouse in the preview area and choose **Open...** from the contextual menu; then select the file containing the picture to be displayed. You can use any picture format.
  The selected picture is displayed immediately in the preview area and in the editor window.
- To change the display format of the picture, choose a value from the **Picture Format** menu. The formats provided are the standard picture display formats of 4D.
- To delete a custom picture, click on the **Clear** button or click with **right button** of the mouse in the preview area and choose **Clear...** from the contextual menu.

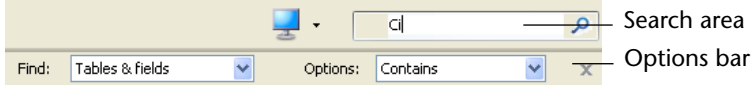
- **Fill Color**: You can change the color used for the background of the editor window. To do this, click in the color selection area and choose a color from the selection menu.

Find in Structure

4D lets you carry out searches in the Structure editor window. Searches can be among the following elements:

- table names and/or field names
- table numbers.

To do a search, enter the character string or table number to be searched for in the Find in Structure area of the Structure editor. Entering a value in this area displays an options bar below it that can be used to specify the scope and type of search desired:



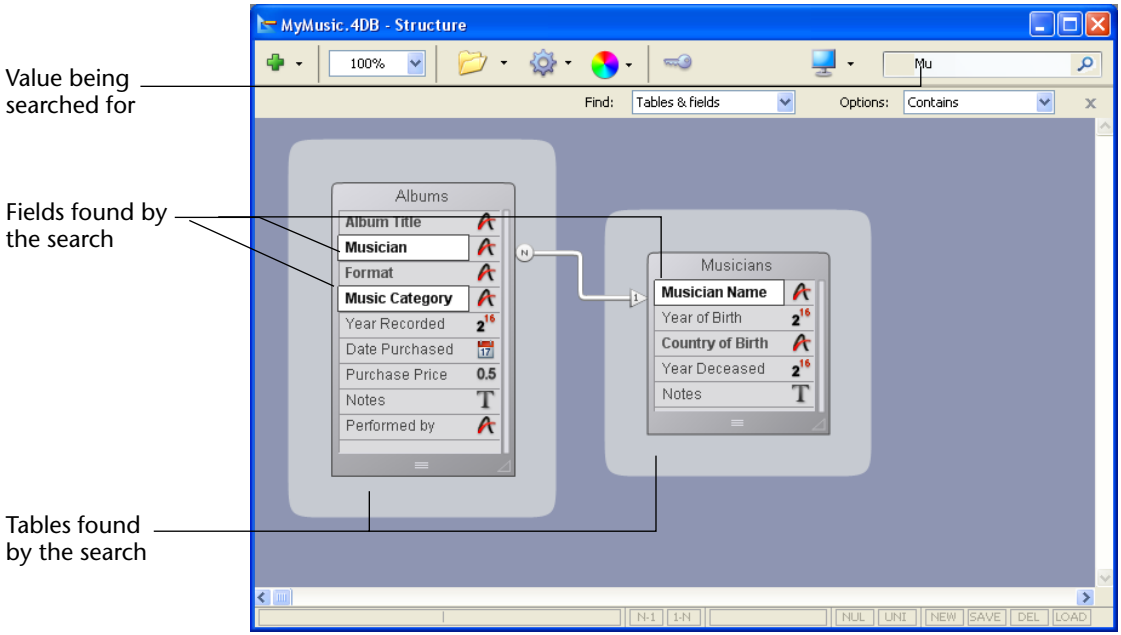
- The Find menu can be used to set the scope of the search (tables and fields or tables only):

- The Options menu can be used to set the type of search:

  - *Contains* (default): searching for “le” will find “Table,” “Letter,” “Elements,” etc.

- *Starts with*: searching for “pa” will find “paper”, “paid” but not “repair.”
- *Number*: searching for “2” will find table number 2, 12, 20, 21, etc.

Searches are carried out in real time, as you enter the values. Tables and/or fields found by the search are “lit up.”:



If nothing is found, the search area turns red:

To exit the “find” mode, click on the X button in the search options bar or delete all the characters in the search area.

Creating and Modifying a Table

You can create tables at any time. 4D names the first table [Table_1] and then names additional tables sequentially, up to [Table_32767]. You can rename the tables to suit your purposes. See “Renaming a Table” on page 258 for more information about naming tables.

You can remove obsolete tables or tables created by mistake. This deletion can be permanent or not (see the "Deleting a Table" paragraph on page 263). Be careful, if the deletion is permanent, any data stored in the table are also deleted. You can also remove a table from the editors of the Application mode by making it invisible (see the "Invisible" paragraph on page 261).

Creating a Table

You can create a new table either directly in the Structure window, or via a dialog box. Direct creation is faster; but creation using the dialog box lets you specify certain parameters of the table and cancel the operation when necessary.
You can also create tables using copy-paste.

► To create a new table directly:

1 Right-click an empty area of the Structure editor window, then choose Add Table... from the contextual menu.
OR
In the add objects menu of the Structure editor tool bar, choose the Table option:

4D creates the table directly in the Structure editor. You can then change its name, add fields, etc. (see the following paragraphs).

Note If you want to put the table in a specific folder, you will need to use the Explorer (see the "Home Page" paragraph on page 103).

► To create a new table using the "New table" dialog box:

1 Choose New > Table... in the File menu (or click on the *New* button of the 4D tool bar).
OR
On the Tables page of the Explorer, click on the add button .

The following dialog box appears:

2 **(optional) Modify the name of the table you want to create.**

For more information about naming rules, please refer to the "Rules for Naming Tables and Fields" paragraph on page 287.

3 **(optional) Specify the folder in which you want to place the new table.**

By default, the table is placed at the highest level of the folder hierarchy ("Top Level"). For more information about object folders, refer to the "Home Page" paragraph on page 103.

4 **Click OK.**

If you want to cancel the operation, click the **Cancel** button.

4D creates a new table image. It becomes the selected table image in the Structure editor window.

► To create a new table by copy-paste:

1 **Select the table(s) to be duplicated and choose the standard "copy" command (4D Edit menu, contextual menu or using the Ctrl+C/Command+C shortcut).**

2 **Then choose the Paste command either in the Edit menu, in the contextual menu or using the Ctrl+V/Command+V shortcut.**

The table is pasted with all its fields. It is renamed by default "Copy(X)_of_*TableName*," where *TableName* is the original name of the table and *X* is the number of copies of the table.

Renaming a Table

You can rename tables at any time. If you have used the old table name in a method, 4D automatically changes it to the new name *provided the method is closed*. If the method is open, you must make the changes yourself.
Each table name must be unique in the database.

You can rename a table directly in the Structure editor, in the Explorer, or using the Inspector palette.

► To rename a table directly (Structure editor or Tables page of the Explorer):

1 **Click twice on the table name (if the table is already selected, click once).**

It switches to editing mode:

Explorer

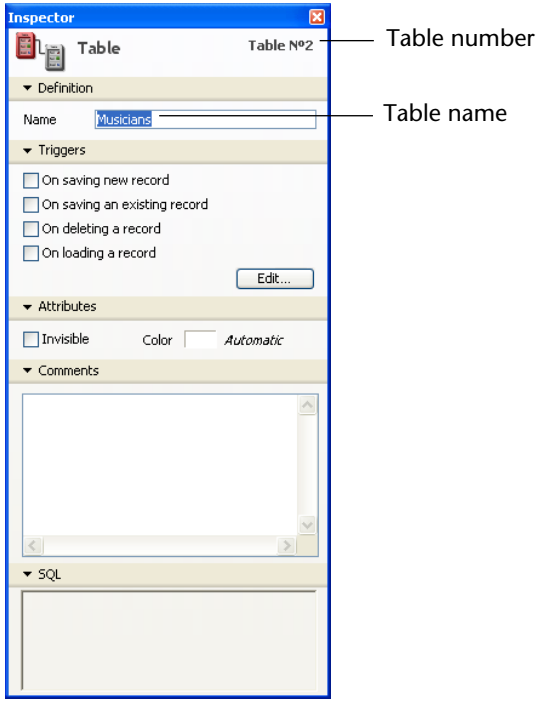
2 **Enter the new name and click outside of the table.**

To find out the rules for naming tables, please refer to "Rules for Naming Tables and Fields" paragraph on page 287.

► To rename a table using the Inspector palette of the Structure editor:

1 **Double-click on the table name.**
OR
Right-click on the table image, then select Table Properties... in the contextual menu.
OR
In the Explorer, double-click on the table name on the Tables page.

4D displays the name and properties of the selected table in the Inspector palette. This palette also indicates the table number:



Note For more information about using the Inspector, please refer to the "Inspector Palette" paragraph on page 246.

If the Inspector palette is already open but is displaying the properties of another table or another object, just select the desired table image and the palette will then display its properties.

2 Enter the new name in the "Name" area.

The new name is applied immediately. To find out the rules for naming tables, please refer to the "Rules for Naming Tables and Fields" paragraph on page 287.

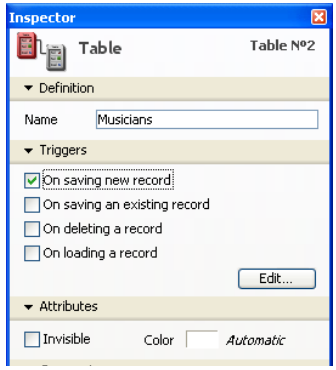
Setting Trigger Events

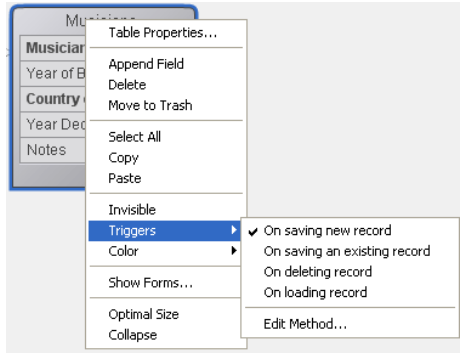
A trigger is a method that runs automatically when certain events related to the table occur. The events are:

- On saving new record,
- On saving an existing record,
- On deleting a record,

- On loading a record.

Trigger events can be set in the Inspector palette or using the contextual menu associated with the table (right-click on the title area of the table):





Check each event for which you want to activate the trigger.

You can access the trigger of a table directly from the Structure editor by clicking on the **Edit...** button in the Inspector palette or by selecting **Edit Method...** in the contextual submenu. For more information about triggers, please refer to the section “Types of 4D Methods” on page 758.

Notes

- You can also create and display triggers with the Methods page of the Explorer.
- The information bar of the Structure editor indicates the triggers that are activated for each table (see the “Toolbar and Information Bar” paragraph on page 244).

The trigger that you enter in the Method editor will only be executed when the selected event(s) are detected.

Setting Table Attributes

You can set the two following attributes for a table: Invisible and Color. They are set in the “Attributes” area of the Inspector palette or using the contextual menu of the tables.

Invisible

This option lets you make a table invisible in the Application mode and for plug-ins. You can activate this option for utility tables or tables that are not being used temporarily.

Making a table invisible allows you to limit the operations that a user can perform on a table by making the table and its fields invisible in all editors and some dialog boxes that appear in the Application environment. It can also not be used by plug-ins.

The editors and dialog boxes concerned in the Application environment include the following:

- All query editors
- Order By editor
- Chart editor
- Label Wizard
- Quick Report editor
- Import and Export dialog boxes
- Apply Formula dialog box.

In each of these editors, the user is unable to see or choose the table or any of its fields. For instance, the user cannot include any fields from an invisible table in a report or label.

Note When using these editors, users have the option of saving their specifications (e.g., the query or sort they created) to disk files. In this case, any specified tables or fields that are subsequently made invisible will may still be used in the operation. In addition, users can type the names of invisible tables and fields in the Apply Formula dialog box.

Invisible tables and fields are displayed in italics in the Structure editor window.

Setting the Color of the Table Image

A color can be attributed to each table. Using colors helps to organize the structure of a large database. For example, you could use one color for all tables that relate to customers and customer records and another color for tables that relate to inventory and inventory records.

Note It is also possible to attribute a color to each field individually (see the "Color" paragraph on page 279) as well as to each relation (see the "Definition" paragraph on page 309).

To set the color of one or more tables, make your selection and then choose the color using:

- The **Color**  button of the Structure editor tool bar,
- The **Color** command in the contextual menu of the tables,
- The **Color** option in the Inspector palette.

The **Automatic** option can be used toa pply the standard original color to the table.

Comments

The Comments area of the Inspector palette lets you store additional information about the table. These comments are available for all the developers.

Note that each field and each relation has its own comments area .

SQL

The SQL area of the Inspector palette indicates whether the name of the table respects the rules regarding SQL nomenclature (for example, unlike 4D, SQL does not allow a field name to contain spaces).

Deleting a Table

You can delete tables from your database.

This operation can be carried out via the SQL engine of 4D or via the Structure editor.

For more information about using SQL statements in 4D, please refer to the *4D SQL Reference* manual.

There are two ways of deleting tables in 4D: *permanent* deletion (the table and its data are actually removed from the database) and *non-permanent* deletion (the table is simply put into the Trash and may be recovered subsequently).

Permanent Deletion

► To permanently delete one or more database tables from the Structure editor:

1 **Select the table(s) to be deleted then choose Clear from the Edit menu of 4D.**
OR
Choose Delete from the contextual menu of the table.

A warning dialog box will ask you to confirm the operation:

If you click **OK**, 4D will carry out the following operations:

- The table or field is permanently deleted from the structure. All the data associated with the table or the field are deleted permanently from the data file.
- Any trigger method associated with the table is deleted.
- The table forms associated with the table are transformed into project forms and are placed in the Trash of the Explorer.

Non-permanent Deletion

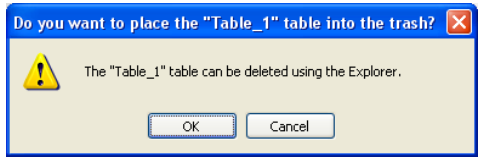
Non-permenent deletion of a table consists in putting it in the Trash of the database (which can be accessed via the Explorer), just like the file deletion mechanisms implemented by Windows and Mac OS.

The table then no longer appears in the 4D editors and its contents become inaccessible but it can still be recovered so long as the Trash is not emptied.

► To put one or more tables into the Trash from the Structure editor:

1 **Select the table(s) to be deleted then press the Delete or Backspace key.**
OR
Choose the Move to Trash command from the contextual menu of the table.

The following dialog box will appear:



If you click **OK**, the table is put into the Trash. It can be recovered at any time. For more information, please refer to the "Trash Page" paragraph on page 129.

Numbers of Deleted Tables

When a table is deleted, the other tables of the database are not renumbered, so as to avoid endangering the stability of the database. It is therefore possible, for example, to have a database with three tables numbered 2, 4 and 5.

Note that the numbers of permanently deleted tables are reused when new tables are created subsequently.

Creating Fields and Setting Field Properties

For each table, you need to create the fields that hold the data you want to store and manage.

When you create a field, you assign it a field type that describes the kind of information that will be stored in it. 4D uses the field type to perform different kinds of operations on the contents of the field. For example, if a field will contain a date, you will want to create it with a Date field type. Subsequently, 4D can compute date values, such as length of service or qualification for benefits.

In addition, 4D can sort records in chronological order using the dates in this field. Field types are described in detail further along in this chapter.

In addition to the field type, each field in a table can possess various properties. These properties determine conditions for entering, displaying, or modifying data in the fields. They are described further on in this chapter.

After creating a field, you can always later change its type or properties, or even delete it. If you create an unnecessary field, you can also make it invisible for users by selecting the Invisible property.

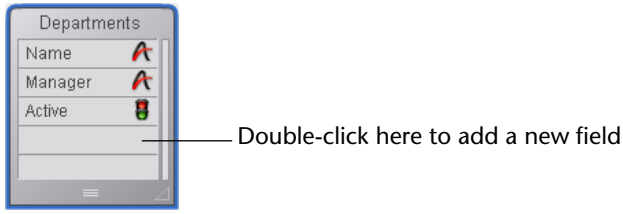
Creating a Field

You can create up to 32,767 fields in a table. You can create a new field using standard creation commands or by copy-paste .

► To create a field:

1 Right-click on a table, then select Append Field or Insert Field[1] from the contextual menu.
OR

Double-click on one of the empty rows below the last field name of the table.



OR

Select a table then, in the add objects menu of the Structure editor tool bar, choose the Field option:

The field is then added to the table and 4D selects it. By default, the field is named *Field_X* where X corresponds to the current field number (if fields have been deleted, this number may not correspond to the actual number of fields in the table, see the "Numbers of Deleted Fields" paragraph on page 287).

You can also change its names, set its properties, etc. (see the following paragraphs).

2 **To quickly create additional fields, you can hit the Carriage return as many times as necessary.**

► To create a field using copy-paste:

1 **Select the field(s) to be duplicated and choose the standard "copy" command (4D Edit menu, contextual menu or using the Ctrl+C/Command+C shortcut).**

---

1. The **Insert Field** command appears when you click on an existing field of the table. The field is then created visually at the location of the click. To find out more about the order fields appear in, please refer to the "Reordering Fields" paragraph on page 269.

2 **Select the table where you want to add the new fields and then choose the Paste command either in the Edit menu, in the contextual menu or using the Ctrl+V/Command+V shortcut.**

The field is duplicated with the same type and properties. If a field with the same name already exists in the destination table, it is renamed by default "Copy_of_*FieldName*," where *FieldName* is the original name of the field. An additional number is added if you make several copies of the same field.

Renaming a Field

You can rename fields at any time; 4D will update the names of the fields wherever they are being used (forms, methods, files). Be careful, if methods were open in the Method editor when the field name is changed, they will need to be closed and reopened in order for the name of the field to be updated.

Each field name in a table must be unique.

You can rename a field directly in the Structure editor and in the Explorer, or using the Inspector palette.

► To rename a field directly (Structure editor or Tables page of Explorer):

1 **Click twice on the field name (if the field is already selected, click once).**
OR
(Structure editor only) Select the field to rename and hit the Carriage return.

It switches to editing mode:

Structure editor

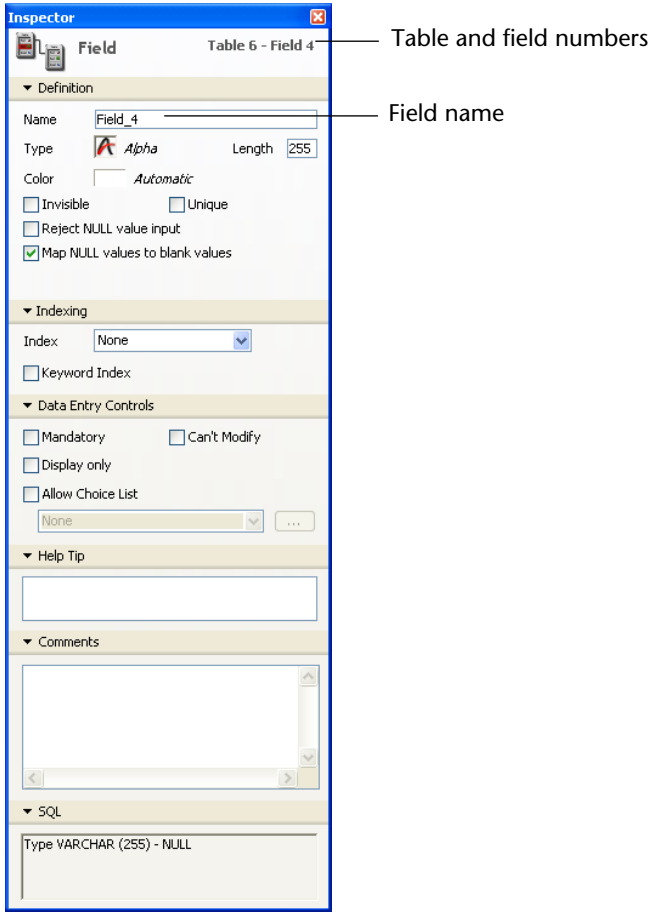
Explorer

2 **Enter the new name and click outside of the field.**

To find out the naming rules for fields, please refer to the "Rules for Naming Tables and Fields" paragraph on page 287.

► To rename a field using the Inspector palette of the Structure editor:

1 **Double-click on the field.**
OR
Right-click on the field, then select Field Properties... from the contextual menu.
OR
In the Explorer, double-click on the field name on the Tables page.
4D displays the name and properties of the selected field in the Inspector palette. The palette also indicates the table and field numbers :



Note For more information about using the Inspector, please refer to the "Inspector Palette" paragraph on page 246.

If the Inspector palette is already open but displaying the properties of another field or another object, select the desired field in order to display its properties in the palette.

2 **Enter the new name in the "Name" area.**

The new name is applied immediately. To find out the rules for naming fields, please refer to the "Rules for Naming Tables and Fields" paragraph on page 287.

Reordering Fields

4D lets you modify the order in which fields appear in the tables of the Structure editor. Changing the order of fields in tables can, for example, make analysis of the structure easier.

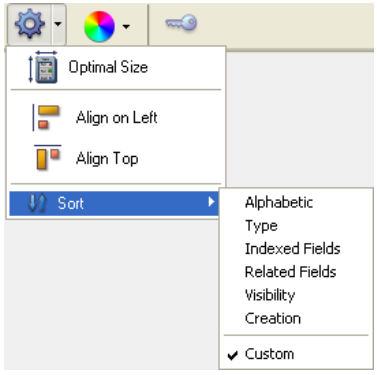
Note that reorganizing fields in the Structure editor does not effect their display in the other editors of the application. In fact, 4D always displays the fields in their order of creation (including when the **Insert Field** command is used) — except in the case of the Explorer, where objects are displayed by alphabetical order.

There are two ways to modify the order of the fields:

- Manually reorganize the contents of a table by simply dragging and dropping its fields. Simply press the **Alt** (Windows) or **Option** (Mac OS) key — the cursor changes into a hand — and move each field wherever you want it:

Custom sorting carried out in this way is memorized and can be applied again if necessary using the **Custom** command of the sort menu (see below).

- Applying a sort criterion. You have several sort criteria available using the **Sort** sub-menu associated with the tools button of the editor:



When you choose an option, it is applied to all the tables selected in the window. This menu is disabled when no table is selected

The menu also indicates the criterion that is currently applied with a checkmark. Several checkmarks are displayed when different criteria are used in the selection of tables.

You can select a sort criteria from among the following:

- **Alphabetic**: Displays the fields by alphanumeric order.
- **Type**: Displays the fields by type (with type names in alphabetical order).
- **Indexed Fields**: Displays the indexed fields at the top.
- **Related Fields**: Displays the primary keys and then the foreign keys.
- **Visibility**: Displays visible fields at the top
- **Creation**: Displays the fields in the creation order (default).
- **Custom**: No specific sorting is applied. This option restores the previous sort carried out manually by drag and drop. If no drag and drop has been carried out previously, this option will have no effect.

Note When you add a field to a table, it is always placed after the last existing field, regardless of the current sort criterion.

Setting Field Types

You must specify a field type for each field. Field types affect how 4D manipulates and stores data in a field and how you enter or display data in forms.

► To modify a field type:

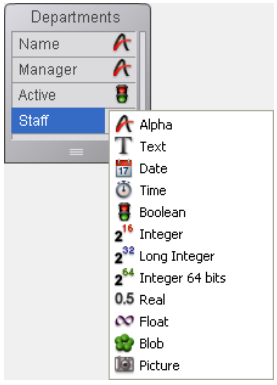
1 Click on the field type icon in the table image and choose the type you want in the associated menu.

OR

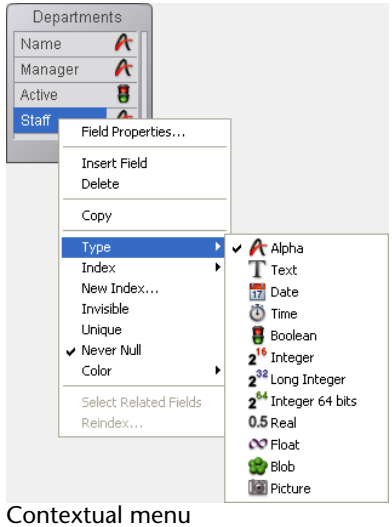
Right-click on the field whose type you want to change and choose the new type from the Type submenu of the contextual menu.

OR

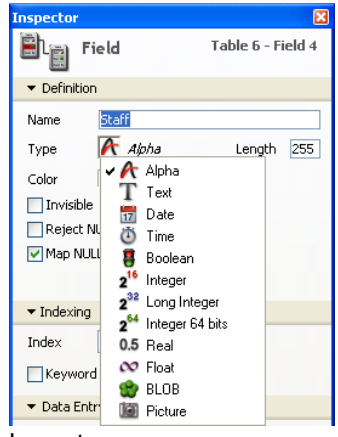
Select the field whose type you want to change and then click in the Type area of the Inspector palette in order to display the types menu.



Direct editing



Contextual menu



Inspector

You can change a field type at any time, unless the field has a relation (in this case, you must remove the relation before you can change the field type).

If you change the field type before entering any data into the field, 4D simply changes the field type.

If you change the field type after entering data into the field, 4D converts the data to the new type if possible, when the data is loaded for the first time after the change. Data from a Picture field converted to any other type does not display. Data from a field converted to a Picture field type does not display.

When you convert a field type, 4D retains the field's original value until you modify the record. For example, if a Text field contains text such as "over 10" and you change the field type to Integer, the modified field displays "10." If you change the field back to a Text field without editing the field value, 4D displays "over 10" again.

By default, 4D supports the following field types:

- **Alpha**: Alphanumeric text between 1 and 255 characters.
- **Text**: Text up to 2 GB.
- **Date**: Date between the year 100 and the year 32,767.
- **Time**: Time in hours:minutes:seconds format.
- **Boolean**: A field that can only take the values TRUE or FALSE.
- **Integer**: Number between -32,768 and 32,767.
- **Long integer**: Number in the range of plus or minus 2,147,483,647.
- **Integer 64 bits**(1): Number on 8 bytes included between +/- 2E64,
- **Real**: Floating point number in the range of ±1.7E±308.
- **Float**(1): Floating point number,
- **BLOB**: Any binary object such as a graphic, another application, or any document.
- **Picture**: A PICT picture.

(1) *Be careful, these field types are only used by the SQL engine of 4D. If these fields are used in the 4D language, their values are converted internally into real numbers.*

Some field types can be grouped by family since their properties are similar:

- *string*: generic term indicating an Alpha or Text type,
- *number*: generic term indicating an Integer, Long integer, Integer 64 bits, Real or Float type.

Alpha

An Alpha field contains alphanumeric characters (letters and numbers), punctuation marks, and special characters such as the asterisk (*), percent sign (%), hyphen (-), and so on. Use an Alpha field to contain any information that must be treated as text and does not exceed 255 characters in length.

An Alpha field can be associated with a standard and/or keywords index. For more information about indexing, please refer to the "Management of Indexes" paragraph on page 288.

Alpha is the most common field type. Typically, you use this field type for names, addresses, telephone numbers, postal codes, and so forth. During data entry, an Alpha field accepts any character, number, punctuation mark, or special character.

Zip codes are best placed in an Alpha field for two reasons: Numeric fields do not display leading zeros and some zip codes contain a hyphen. The general rule for deciding between a numeric field type or an alphanumeric field type is make it an alphanumeric field unless it will be used in a numerical calculation or searched or sorted based on numeric values.

You can set the maximum length of an Alpha field to be between 1 and 255 characters long.

You can concatenate two or more Alpha fields. For instance, you might want to join a person's first name and last name for the first line in a label form.

You can do so using a one-line method, such as:

```
FullName:=[Employees]First_Name+" "+[Employees]Last_Name
```

The variable *FullName* can be displayed or printed. You can also extract part of the information for use in another place (extraction of a substring). The substring can be displayed or printed.

Text

A Text field is similar to an Alpha field, except for a few points.

A Text field can hold up to 2 GB of alphanumeric characters. In geneéral, you use a Text field to hold large blocks of text, like comments or descriptions.

For reasons concerning optimization, the contents of a Text field are stored by default outside of records (see the "Stored in the Record" paragraph on page 282). In this case, the field cannot be associated with a standard index. Like Alpha fields, Text fields can be associated with a keyword index. For more information about indexing, please refer to the "Management of Indexes" paragraph on page 288.

In an input form, a Text field can have scroll bars. In a printed report, the Text field area can expand as necessary to print all the information, even if it covers several pages.

During data entry, Text fields provide basic text editing features: scrolling, word wrapping within the area set for the field display, double-clicking to select a word, moving the insertion point with the arrow keys, and standard cut, copy, and paste operations. If it has the **Multiline** option (see the "Multiline" paragraph on page 284), a Text field accepts a carriage return during data entry to create a new paragraph (an Alpha field does not).

You can paste text into Text fields, including text from word processors.

Note Another way to store text with a record is to use the 4D Write plug-in. With 4D Write, you can use different font attributes, paragraph alignments, and other word processing features that are not available in standard Text fields.
As with all 4D plug-ins, the 4D Write area must be placed in a field of the BLOB type — and not the Text type. For more information about using 4D Write, refer to its documentation.

Date

Use a Date field to store date values such as Start Date, Date Purchased, Birthdate, and so on. A Date field can store any date value (month, day, year) entered in a *MM/DD/YYYY* format between the year 100 and the year 32,767.

Note In the United States, dates are specified in the month/day/year (*MM/DD/YYYY)* format. Other countries use different formats such as *DD/MM/YYYY* for British systems and *YY/MM/DD* for Swedish systems. 4th Dimension will store the date based on the format specified by the operating system of your computer.

Time

Use a Time field type to manage times such as Current Time, Meeting Time, Billed Time, and so on. A Time field can store any time value entered in *HH:MM:SS* format.

Boolean

Boolean fields (sometimes called logical fields) contain TRUE or FALSE values.

You can format a Boolean field as either a check box or as a pair of radio buttons. A check box that contains a check is TRUE; empty, it is FALSE. Either the first radio button is selected (TRUE), or the second button is selected (FALSE).

You should name a Boolean field so that you can ask the question, "Is *field name* true?" This question is useful for searching because during a search, 4D looks for a TRUE and FALSE value in a Boolean field. For example, you might want to name a field "Male" instead of "Sex." Your search condition can then be written "Male is equal to True," instead of "Sex is equal to True."

Integer

Use an Integer field type for any field that stores whole numbers, that is, numbers without decimals (record number, invoice number, and so on). Integer fields can contain whole numbers between -32,768 and 32,767.

Long Integer

Use a Long Integer field type for any field that stores whole numbers that are too large for an Integer field. They can contain whole numbers (no decimal) between ±2,147,483,647.

Integer 64 bits

This type of field can be used to store whole numbers on 8 bytes which allows the handling of very large numbers, included between +/- 2E64.

Note Be careful, this type of field is only used by the SQL engine of 4D. If this field is used in the 4D language, its value is converted internally into a real number.

Real

A Real field stores real numbers, that is, decimal numbers (price, salary, expenses, and so on). Real number fields can hold any number in the range of ±1.7E±308.

Numeric display formats are automatically based on system regional parameters. 4D replaces the "," and "." characters in numeric display formats by, respectively, the thousands separator and the decimal separator specified in the operating system.

Note In previous versions of 4D, numeric display formats did not take the system regional parameters into account. In converted databases, the mechanism based on the operating system must be explicitly enabled via the "**Use system settings in numeric formats**" option (see the "Design Compatibility" paragraph on page 173).

Float

This field type is used to store floating point numbers. These types of numbers store real values without any loss of accuracy.

Note Be careful, this type of field is only used by the SQL engine of 4D. If this field is used in the 4D language, its value is converted internally into a real number.

BLOB

Blob (**B**inary **L**arge **Ob**ject) fields store binary documents of any kind. For example, you can store documents created by other applications, scanned pictures, or other applications. A BLOB can be as large as 2 gigabytes. When you are working with a record that contains a BLOB field, the entire BLOB is loaded into memory. You can use a BLOB field to store entire desktop documents within your database. You can also write the contents of a BLOB field to a desktop document. For example, you can use a BLOB field in a document management system that stores documents in the database and delivers them to users upon request.

You use BLOB commands in 4D's language to manage BLOB fields. Use the DOCUMENT TO BLOB and BLOB TO DOCUMENT commands to read and write documents to and from BLOB fields. The commands COMPRESS BLOB, EXPAND BLOB, and BLOB PROPERTIES let you work with compressed BLOBs. For more information about working with BLOBs, see the section on BLOBs in the *4D Language Reference* manual.

For reasons concerning optimization, the contents of BLOB fields are stored outside of records. BLOBs are only loaded when necessary, for example once the record being searched for has been found.

The contents of a BLOB field are not displayed on-screen since a BLOB can represent any type of data.

Picture

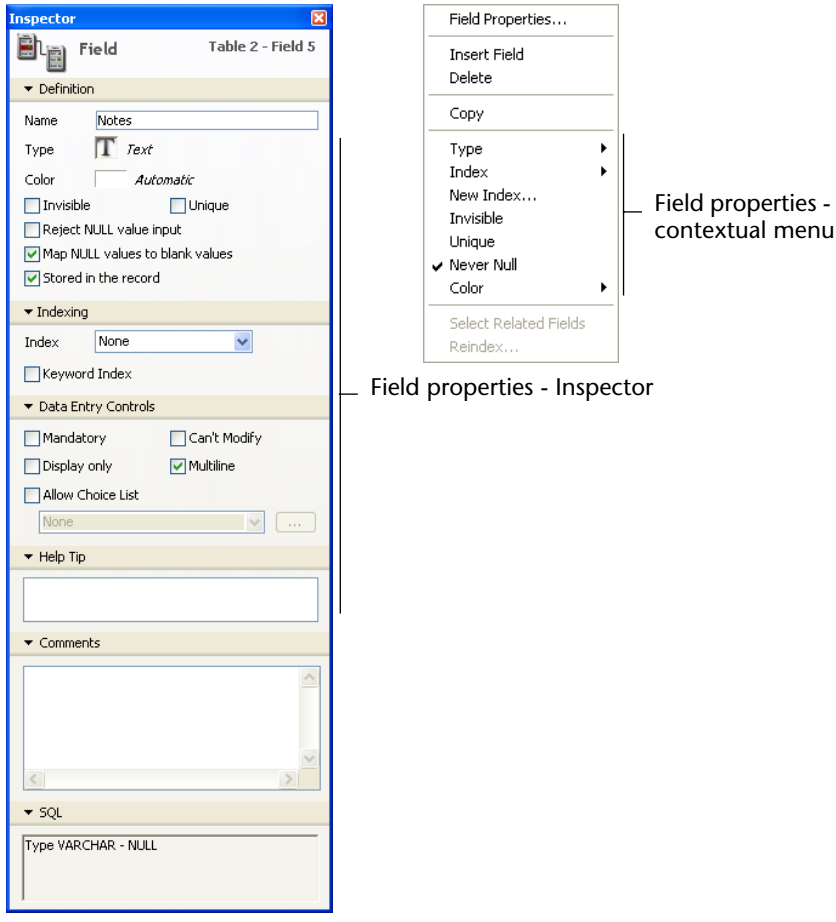
Picture fields are used to store digitized photographs, diagrams, maps, and illustrations created using a graphics application. The pictures are kept in their native format. Some graphic applications store extra information with pictures that may provide special instructions for output devices such as a PostScript™ printer. This information "tags along" when the picture is copied or pasted into a Picture field and is used by 4D when printing the picture to an appropriate output device.

For reasons concerning optimization, the contents of Picture fields are stored outside of records. Pictures are only loaded when necessary, for example once the records being searched for has been found.

Setting Field Properties

Field properties determine its appearance as well as its data entry, display, editing and data storage conditions. The properties of each field in a table can be set individually.

Field properties can be specified in the Inspector palette or, for certain ones, in the contextual menu that appears when you right-click on a field:



Notes

- When you modify the properties of a field, the changes generally do not affect any data already entered in this field — except for indexing properties. For example, if you choose the Unique attribute, only entries made after this change will be checked for uniqueness; any values entered before the modification may contain duplicate values.
- For more information about using the Inspector, please refer to the "Inspector Palette" paragraph on page 246.

Color

You can assign a color to each field. Colors can be used to distinguish fields according to their role or attributes. For example, you can use one color for unique fields and another for mandatory ones.

Note The field color set in the Structure editor has no effect on the color of fields displayed in forms.

The color set for a field will be applied to the field name.

Note It is also possible to assign a color individually to each table (see the "Setting the Color of the Table Image" paragraph on page 262) and to each relation (see the "Definition" paragraph on page 309).

To set the color of one or more fields, select them and choose a color using:

- The **Color**  button in the tool bar of the Structure editor,
- The **Color** command in the contextual menu of the fields,
- The **Color** option in the Inspector palette.

The **Automatic** option can be used to apply the standard original color of the field.

Invisible

You can make a field invisible in the Application environment by selecting the **Invisible** property for this field. The Invisible attribute hides the field from the user. A field with this attribute does not appear in any standard 4D editors and dialog boxes that appear in the Application environment. In addition, it cannot be used by plug-ins.

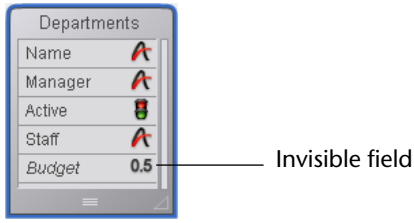
The following editors and dialog boxes in the Application environment are concerned :

- All query editors
- Order By editor
- Chart editor
- Label Wizard
- Quick Report editor
- Import and Export dialog boxes
- Apply Formula dialog box.

In each of these places, the user is unable to see or choose the field. For instance, the user cannot choose an invisible field for a report created with the Quick Report editor.

Note When using the editors, users have the option of saving their specifications (e.g., the query or sort they created) to disk files. In this case, any fields specified that are subsequently declared invisible will still be used in the operation. In addition, users can type the names of invisible fields in the Apply Formula dialog box.

Invisible fields are displayed in italics in the Structure editor window.



Unique

Use the Unique attribute when you want to be certain that each record has a different (unique) value in this field. The Unique attribute should be used for the field that uniquely identifies each record in the table. The Unique attribute is useful to validate fields that store Employee numbers, Social Security numbers, Purchase Order numbers, and so on.

If you want to set the Unique attribute for a field, you must first make it an indexed field. The Unique attribute is disabled unless the Indexed attribute is set.

The Unique attribute prevents duplication of empty values as well as actual entries. An empty field cannot be duplicated in another record.

Note The information bar of the Structure editor indicates whether the Unique property is checked for a field (see the "Toolbar and Information Bar" paragraph on page 244).

Reject NULL Value Input

The **Reject NULL value input** property is used to prevent the storage of NULL values.

When this attribute is checked for a field, it will not be possible to store the NULL value in this field. This low-level property corresponds exactly to the NOT NULL attribute of SQL. Generally, if you want to be able to use NULL values in your 4D database, it is recommended to use exclusively the SQL language of 4D.

Note In 4D, fields can also have the "Mandatory" attribute. The two concepts are similar but their scope is different: the "Mandatory" attribute is a data entry control, whereas the "Reject NULL value input" attribute works at the level of the database engine.

If a field having this attribute receives a NULL value, an error will be generated.

Map NULL Values to Blank Values

This property make the processing of "blank" values and NULL values consisten for the field via the 4D language.
For compatibility reasons, NULL values stored in 4D database tables are automatically converted into default values when being manipulated via the 4D language. For example, in the case of the following statement:

```
myAlphavar:=[mytable]MyAlphafield
```

... if the *MyAlphafield* field contains a NULL value, the *myAlphavar* variable will contain "" (empty string).

The default values depend on the data type:

- For Alpha and Text data types: ""
- For Real, Integer and Long Integer data types: 0
- For the Date data type: "00/00/00"
- For the Time data type: "00:00:00"
- For the Boolean data type: False
- For the Picture data type: Empty picture
- For the Blob data type: Empty blob

On the other hand, this mechanism in principle does not apply to processing carried out at the level of the 4D database engine, such as queries. In fact, searching for a "blank" value (for example *myvalue=0*) will not find records storing the NULL value, and vice versa. When both types of values (default values and NULL) are present in the records for the same field, certain processing may be altered or require additional code. To avoid these inconveniences, the **Map NULL values to blank values** option can be used to standardize all the processing in the 4D language. This property can be used to extend the principle of using default values to all processing. Fields containing NULL values will be systematically considered as containing default values.

This property is taken into account at a very low level of the database engine. It acts more particularly on the Is field value Null command.

Stored in the Record

This option is only available for Text type fields. It lets you "force" the storage of text data in the records themselves. In fact, for reasons concerning optimization, the values of Text fields are stored by default outside of records (see the "Text" paragraph on page 273).

This possibility is necessary in particular if you wish to use "standard" indexes (B-Tree, etc.) with a Text field since these are not compatible with data storage outside of records. The index selection menu is hidden when Text data are stored outside of records.

QuickTime Compression

You can apply QuickTime® compression to picture fields via the **Compression** option and the associated **Options** button. Selecting this attribute lowers the disk space required for storing pictures. This option is only taken into account for pictures pasted by users into forms.
Under Windows, you must have at least version 4 of QuickTime.

Compatibility note

This option has been kept for compatibility reasons with previous versions of 4D. It does not allow you to take advantage of native picture management in 4D.

Indexing

The **Index** property is available for all field types except for BLOBs and Pictures. The **Keyword Index** property is available for fields of the Alpha and Text types.

Using indexes helps accelerate processing and searches among the data.

Managing indexes is detailed in the "Management of Indexes" paragraph on page 288.

Mandatory

When the **Mandatory** attribute is set for a field, the user must enter a value in that field during data entry. 4D does not accept a record that contains an empty mandatory field. You would set the Mandatory attribute for a field that contains essential information for your database. The field that uniquely identifies each record is a good candidate for the Mandatory attribute. Social Security numbers, invoice numbers, certain dates, or employee numbers might need to have the Mandatory attribute set to protect the integrity of the records.

You can also set the Mandatory attribute for a field in a particular form. If you select the Mandatory attribute in the Structure editor, you cannot deselect it on a particular form. However, you can apply the Mandatory attribute on a form to a field that does *not* have this attribute in the Structure editor. For information about setting the Mandatory attribute for a field in a form, see the section "Setting the Enterable and Mandatory Attributes" on page 490.

Can't Modify

If the Can't Modify attribute is set for a field, 4D validates the value initially entered in the field, but does not allow the user to modify the value after the record has been saved. The user can edit an entry in such a field only during the initial creation of the record, before the record is validated. Once the user saves the record, the value in the field not editable. The value can then only be modified by a method or in the Design environment after you have first removed this option.

Use Can't Modify for fields that must provide an audit trail such as Date Received, Date Paid, and so on. The Can't Modify attribute is often used for the field that uniquely identifies each record in the table.

Note This attribute only works on fields displayed in an input form in Page mode. In other cases (entry in list, entry in a subform in either List or Page mode), the value of the field can still be modified.

Display Only

The user cannot enter values from the keyboard into a field that has the Display Only attribute set. You must use a default value for such a field or write a method that inserts a value in the field. A field with the Display Only attribute is useful for displaying values that you do not want database users to modify, such as calculated totals or a sequence number assigned by a method.

You can also make any field non-enterable on a particular form. For information about making a field non-enterable, see the section "Setting the Enterable and Mandatory Attributes" on page 490.

Multiline

This option is only available for Text type fields. When it is checked, the Text field is automatically configured, in forms created subsequently, so as to contain several lines of text. Its default characteristics are as follows:

- Height corresponding to several lines,
- Horizontal scrollbar,
- During execution, a carriage return causes a line break.

When this option is not checked, the default appearance of Text fields in forms is the same as that of Alpha fields: a single line's height and no scrollbar; a carriage return will move you to the next field of the form.

It is possible to change the default field appearance at any time using the Form editor.

Allow Choice List

Use the **Allow Choice List** attribute if you want to display a choice list for entering information in the field. To use this attribute, you first need to create the choice list using the Lists editor.

Use the Allow Choice List attribute when you want to standardize entries in the field and avoid misspellings. Use a choice list for a field that has a limited number of valid entries or a limited number of usual entries. Using a choice list does not necessarily prevent the user from typing a different value (one that does not appear in the choice list). For more information about choice lists, see the chapter "Lists" on page 903.

You can also assign a choice list to a field on a particular form. However, when you assign a choice list only on a form, the list is not displayed in other editors and dialog boxes, such as the Query editor. For information about using a choice list in a form, see the section "Using Choice Lists" on page 492.

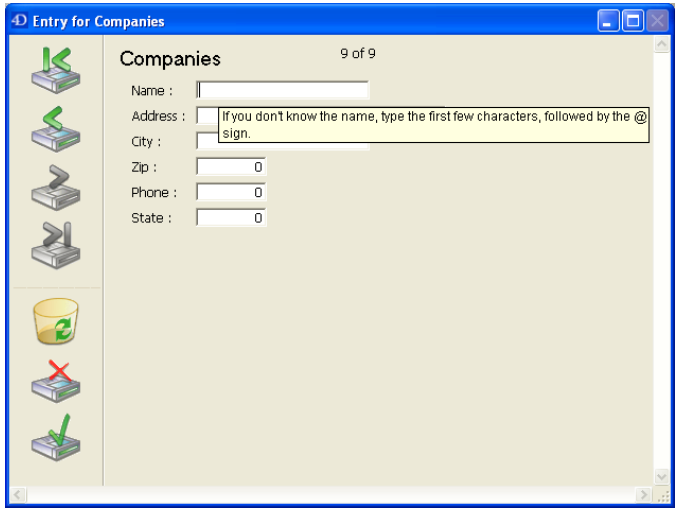
Note It is also possible to set choices lists of **required** values and **excluded** values at the form level.

When you check the **Allow Choice List** option, the associated scrolldown menu is enabled. You can then select an existing choice list that you want to assign to the field or you can click on [...] to directly access the Lists editor.

Help Tips

You can provide users with additional information about a field by adding a help tip to the field. When a tip is entered in the Help Tips area, it appears next to the field whenever a user places the pointer over the field in any form in which the field is included. A Help Tip is displayed on all platforms.

When the user places the mouse pointer over the field, the tip appears below the field, as shown below.



You can also use a help tip for a field in a particular form. If you assign a help tip only to a certain form, it will not appear in the other forms. For more information about using help tips with forms, refer to the "Adding Help Tips to a Field or Object" paragraph on page 517.

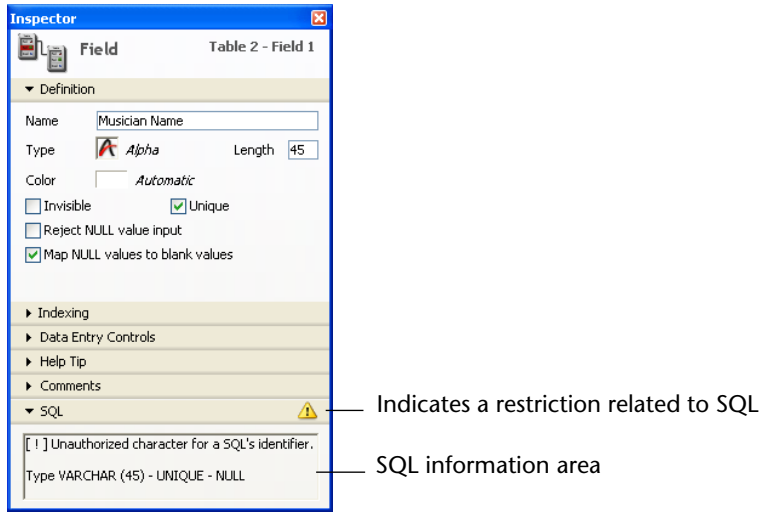
Comments

The Comments area of the Inspector palette can be used to store additional information about the field. These comments are available to all the developers.

Note that each table and relation has its own comments area.

SQL

The SQL area of the Inspector palette provides various useful information about the field from the perspective of its use via the SQL language:



This area indicates more particularly whether the name of the field respects the rules regarding SQL nomenclature (for example, unlike 4D, SQL does not allow a field name to contain spaces).

For fields, the SQL information area also indicates their SQL attributes (type and properties).

Deleting a Field

You can delete fields from your tables. However, this cannot be undone.

This operation can be carried out either via the SQL engine of 4D or via the Structure editor.

For more information about using SQL statements in 4D, please refer to the *4D SQL Reference* manual.

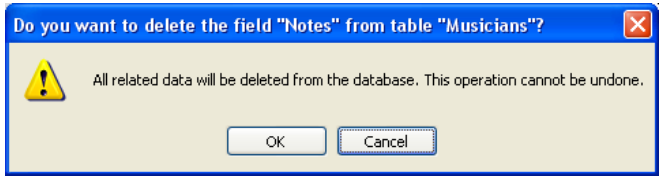
► To delete one or more fields in the database using the Structure editor:

1 Select the field(s) to be deleted.

You can select fields from several different tables at the same time.

2 **Choose Clear in the Edit menu of 4D.**
OR
Choose Delete in the contextual menu of the table.
OR
Press the Delete or Backspace key.

A warning dialog box appears so that you can confirm the operation:



If you click on **OK**, the field(s) will be deleted from the structure. All the data associated with the field(s) are removed from the data file.

Numbers of Deleted Fields

When a field is deleted, the other fields of the database are not renumbered, so as to avoid endangering the stability of the database. It is therefore possible, for example, to have a table with four fields numbered 1,4, 6 and 8.

Note that the numbers of permanently deleted fields are reused when new fields are created subsequently.

Rules for Naming Tables and Fields

Field and tables names must respect the following rules:

- The name can contain up to 31 characters.
- It must start with a letter of the alphabet.
- It may contain any combination of letters, numbers, spaces and underscores.
- Commas, slashes, quotation marks, colons and most other punctuation marks are not allowed.
- 4D truncates table names that exceed 31 characters and removes any spaces from the beginning or end of the name.
- Do not use the same name for two visible objects. If you use the same name twice, a warning dialog box will inform you that another visible object already has the same name and that the entry is not possible.

- Do not use reserved names for naming a table or field. Reserved names include command names (Date, Time, etc), keywords (If, For, etc.) and constants.

Note Additional rules must be respected when objects must be handled via SQL: only the characters *_0123456789abcdefghijklmnopqrstuvwxyz* are accepted, and the name must not include any SQL keywords (command, attribute, etc.). The "SQL" area of the Inspector palette will warn you if a name does not respect an SQL rule (see the "SQL" paragraph on page 263 et page 319).

Tip Although it is possible to insert spaces in object names, entering a name with no spaces will allow you, in the Method editor, to select the object by double-clicking directly on it. It is thus generally recommended to use an underscore instead of a space.

Management of Indexes

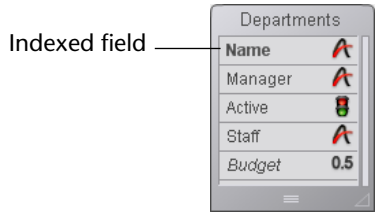
You can associate indexes fields that you frequently use for searching and sorting. For example, you might index Last Name, Company name, or Product name if you plan to search for specific records or sort the records by these fields. You also use this property for fields that establish relations between tables. For more information about this, please refer to the section "Relating Tables" on page 295.

When an index is associated with a field, 4D creates an internal index table for the field. This table allows 4D to perform rapid searches and sorts on the field. When searching or sorting on an unindexed field, 4D moves through data sequentially, examining each record in order. Indexing allows 4D to search and sort without going through every record.

You can index fields of the Alpha, Text, Date, Time, Boolean, Integer, Long integer, Integer 64 bits, Real and Float type. As you add and delete records, 4D automatically updates its index table. If you create an index for a field that already exists, 4D automatically indexes the existing data. You can specify as many indexed fields as you want.

Do not index every field. An index increases the size of the database, using more space on disk. Using many indexes also increases the time needed to save a record since 4D updates the index table with each record validation.

Indexed fields are displayed in bold type in the Structure editor.



Types of Indexes

4D provides different types of indexes. Choosing between the different types is generally based on the result expected and the type of data present in the field.

There are three main types of indexes:

- **Standard indexes**: These are single-field indexes used to accelerate standard database operations (searches and sorts). 4D lets you choose the internal architecture of this type of index: B-Tree or Cluster B-Tree.
- **Composite indexes**: This index stores the combined values of two or more fields that are often searched for together, for example *LastName+FirstName*.
- **Keyword indexes**: These indexes are only available for Alpha and Text type fields. They are intended for facilitate fast searching inside text.

Standard Indexes

A standard index is intended to accelerate database operations (a standard index refers to a generic index as opposed to a keyword or composite index). 4D offers two types of architectures for standard indexes: **B-Tree** and **Cluster B-Tree**.

- **B-tree**: Standard B-Tree type index. This multipurpose index type meets most indexing requirements.
- **Cluster B-tree**: B-Tree type index using clusters. This architecture is more efficient when the index does not contain a large number of keys, i.e. when the same values occur frequently in the data.

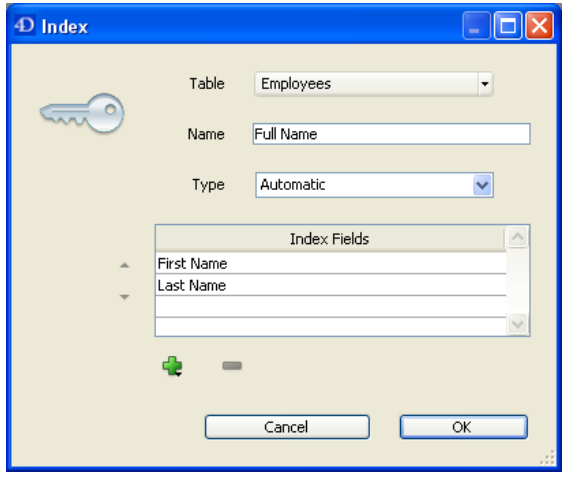
When you choose the index architecture, 4D also provides the **Automatic** option. In this case, 4D automatically selects the architecture according to the type of data concerned.

Composite Indexes

Composite indexes store the combined values of two or more fields fo each entry. The classic example is a composite index based on the FirstName+LastName fields. Searching for “Peter Smith” will therefore be optimized compared with a standard search (searching for “Smith” then searching for “Peter”).

4D automatically takes advantage of composite indexes during queries and sorts. For example, if a composite index “City+ZipCode” exists, it will be used in the case of a query of the type “*lastname=carter* and *city=new york* and *zipcode =102@*”.

Composite indexes can only be created using the index creation dialog box:



For a detailed description of this dialog box, please refer to the “Creating an Index” paragraph on page 292.

Note The index creation dialog box can also be used to create standard or keyword indexes.

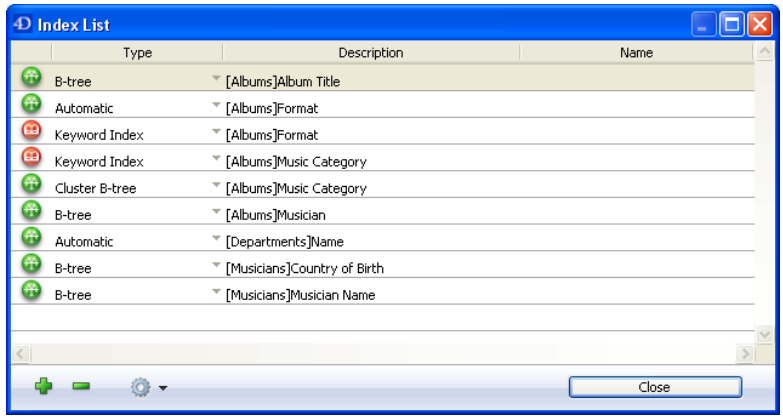
Keywords Index

It is possible to use a specific type of index with Alpha and Text fields: a *keyword index*. When you associate this type of index with a field, the text stored in the field will be indexed word by word. All the words will be indexed even if they have only 1 or 2 characters. This type of index will accelerate subsequent keyword searches among text fields in a dramatic manner (for more information about keyword searches, please refer to the “Comparison Operators” section of the 4D *Language Reference* manual).

Note It is possible to associate both a standard index and a keyword index with Alpha and Text fields (when stored in the records). 4D will use the appropriate index depending on the context.

Index List

The  button of the toolbar in the Structure editor displays the Index List window. This window displays the list of all the indexes of the structure, regardless of their type:



Note For more information about index types, please refer to the "Types of Indexes" paragraph on page 289.

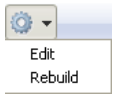
The Index List can be used to view the main properties of the indexes:

- **Type**: Index type. Each type of index (B-tree, Cluster B-Tree, keyword) is depicted with a different icon.
  It is possible to modify the index type from the Index explorer by clicking on the inverted triangle and selecting a value in the pop-up menu.
- **Description**: Table and field(s) of index. For a composite index, this list contains all the fields of the index.
- **Name**: Index name. This property is used in particular by the language commands. You can change or add an index name by double-clicking in this area.

  The  button displays the index property dialog box.

  The button deletes the selected index (a confirmation dialog box appears). This button can be used more particularly to delete composite indexes.

Two additional commands are available in the menu associated with the tool button (enabled when an index is selected):



- **Edit**: Displays the properties of the selected index in the index property dialg box (see next paragraph). This command has the same effect as double-clicking on a row of the list (except for in the name area).
- **Rebuild**: Can be used to delete and rebuild the selected index. A confirmation dialog box appears when you select this command.

Creating an Index

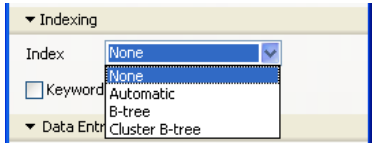
The way an index is created will depend on its type. In addition, you can choose to create an index directly or to use the index creation dialog box.

▶ To create a standard index directly:

1 **Select a field then choose a value from the "Index" menu of the Inspector palette.**
OR
Right-click on the field then select a value from the Index> submenu of the contextual menu.

There are four options available:

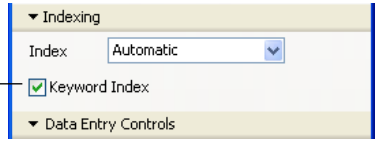
- **B-tree**: Creates a standard B-Tree type index.
- **Cluster B-tree**: Creates a B-Tree type index using clusters.
- **Automatic**: Lets 4D select the architecture depending on the type of data concerned.
- **None**: No index or removal of existing index.



Note For more information about types of indexes, please refer to the "Types of Indexes" paragraph on page 289.

► To create a keyword index directly

1 Select an Alpha or Text type field then check the "Keyword Index" option in the Inspector palette.



Creation of a keyword index

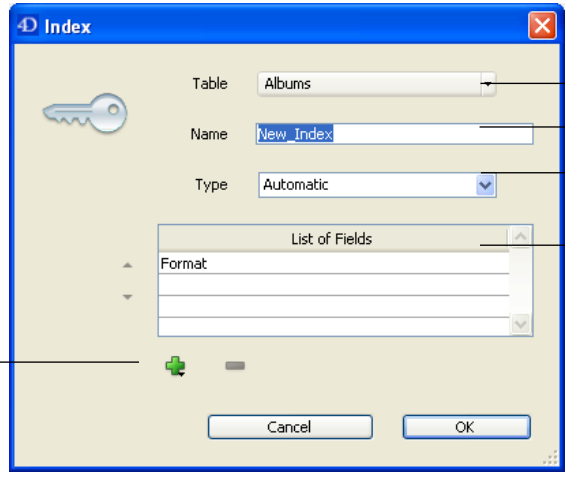
OR
Right-click on a field then select Keywords from the Index> submenu of the contextual menu.

► To create a composite index or any other type of index using the index creation dialog box:

1 Select a field then, using the add objects menu of the Structure editor tool bar, choose the Index option:

OR
Right-click on a field then select New Index from the contextual menu.

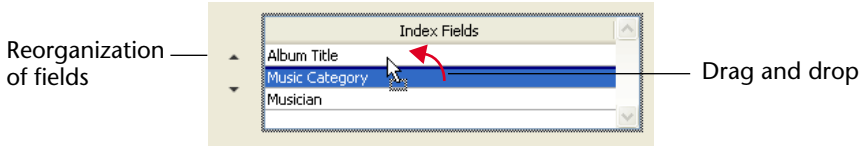
The index configuration dialog box then appears. It contains the following elements:



Index table
Index name
Index type
Index field definition area
Adding and removing index fields

- **Table**: List of all the database tables. Choose the table to which the index will belong from this menu.

- **Name**: Index name entry area. This name is used by the 4D language commands.
- **Type**: Selection menu for type of index to be created. If you keep the "Automatic" option, 4D will automatically choose the index type according to the contents of the field.
- **List of Fields**: This area is used to specify the field(s) associated with the index. It can contain a field by default depending on the current selection in the editor.
  To add a field to the index, click on the button. The list of fields of the selected table is displayed so that you can indicate the field to be added to the index.
  - If you want to create a composite index, add each field to be included in the index successively. Once the list is completed, you can reorder the fields using the arrow buttons or using drag and drop.



  - If you have chosen the "Keyword Index" type, only Alpha or Text fields can be selected. Also in this case, you cannot include only one field in the index.

  To delete a field from the index, select it in the list and click on the button.

  Once the index has been configured, click on **OK** to generate the index.

Deleting an Index in the Structure Editor

You can delete indexes that are no longer useful at any time. This can be carried out directly in the Structure editor or using the List Index window. For more information about the List Index window, please refer to the "Index List" paragraph on page 291.

▶ To delete a standard index:

1 **Select the field associated with the index you want to delete, then choose the None option in the Index menu of the Inspector palette.**
OR
Right-click on the field associated with the index, then choose the None option from the Index> submenu of the contextual menu.

► To delete a keyword index:

1 **Select the field associated with the index you want to delete, then uncheck the "Keyword Index" option in the Inspector palette.**
OR
Right-click on the field associated with the index, then uncheck the Keywords option in the Index> submenu of the contextual menu.

The deletion (and viewing) of a composite index can only be carried out from the List Index window (using the button). For more information about the List Index window, please refer to "Index List" paragraph on page 291.

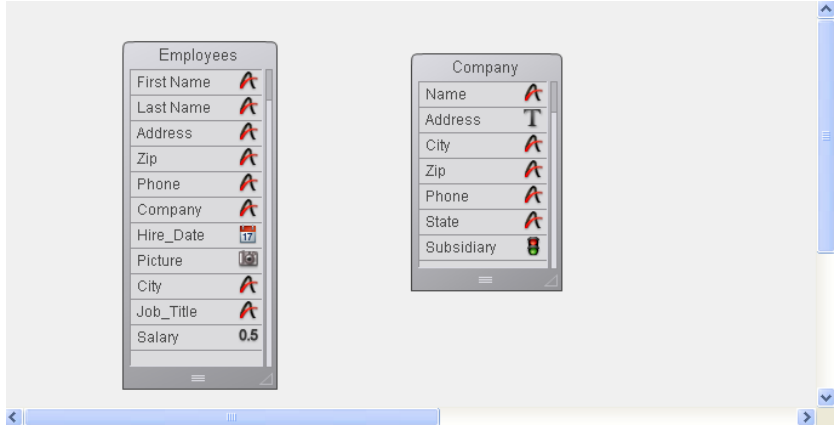
Reindexing a Field

You can reindex a field at any time; in other words, rebuild the index table(s) associated with it, in accordance with the data present. This can be useful in the case of application maintenance.

Reindexing can be carried out either using the **Reindex** command of the contextual menu of the Structure editor (right-click on an indexed field), or using the **Rebuild** command in the Index List window (see the "Index List" paragraph on page 291).

Relating Tables

You will usually need to create structures in which several tables share information. For instance, suppose you create a database to keep track of employees and their companies. The database structure, shown below, contains a [Employees] table for storing employee information and a [Companies] table for storing company information.



Although useful, the information stored in each separate table does not fulfill your information tracking needs. When you are viewing a record from the [Employees] table, you need to be able to view information about the company for which that employee works and when you are viewing a record from the [Companies] table, you need to be able to view information about all the employees who work for that company.

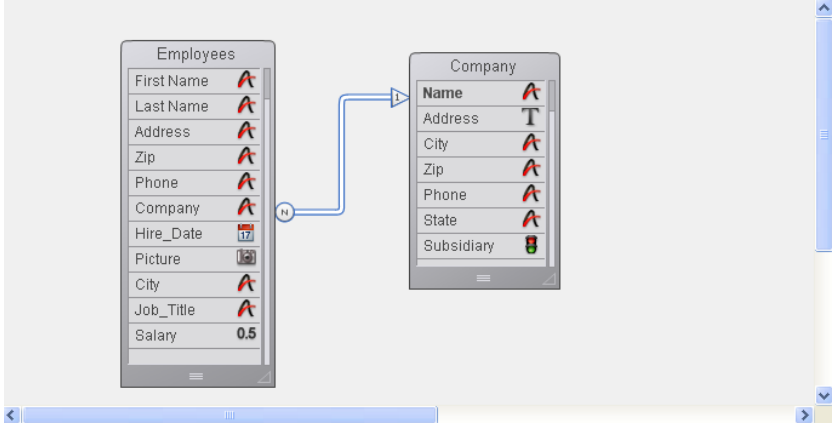
To allow two tables to share information in this way, the tables can be related to each other — in other words, a relation can be established between the data in each table.

In 4D, *table relations* allow data stored in one table to be accessed from another table. Tables that share information by means of a relation are called *related tables*.

Relating tables allows you to do the following:

- Store data efficiently,
- Update data in one place and have the change reflected everywhere the data is used,
- View related information,
- Perform queries and sorts in one table that are based on data in another table,
- Create, modify, or delete records in related tables.

The figure below shows a relation created between the [Employees] table and the [Companies] table in the Structure editor.



The [Employees] table contains one record per employee. The [Companies] table contains one record per company. The relation between the two tables allows you to access, enter, modify, or delete information from both tables. For example:

- When an employee's record is on screen, you can view or modify the corresponding company information — the address, city, state, zip code, and company telephone number.
- When you add a new employee, you can link the employee's record to the appropriate company record (if the company is already entered), or, if the person's employer is not in the database, create the new company record while creating the employee record. For more information, see the section "Entering Data in Related Tables" on page 303.
- For each company, you can view or modify information for each employee in the company — name, title, telephone number, and so forth. You can also add a employee record from within the company record. For more information, see the section "One to Many Options" on page 315.

Related Fields

You are able to display information from related tables by means of the related fields — the fields that connect the two tables in a relation.

The basic purpose for relating tables is to instruct 4D which record or records to make current in one table based on which record is current in the other table. The related tables make use of data in two related fields to identify corresponding records. In the following example, the company name is stored in both the [Employees] table and the [Companies] table.

Related fields

First Name	Last Name	Title	Company
Biff	Davis	Salesperson	Howard Battery Co.
Andy	Venable	Engineer	Howard Battery Co.
Bryan	Pfaff	Secretary	Howard Battery Co.
Kathy	Forbes	Secretary	Howard Battery Co.

Company Name	City	St	Phone
Howard Battery Co.	Arcadia	CA	818-576-2534

The Company field in the [Employees] table and the Name field in the [Companies] table relate the two tables. The Name field in the [Companies] table is the *primary key field* for [Companies]. It uniquely identifies each company record. A primary key should have the Indexed and Unique attributes. The Company field in the [Employees] table is a *foreign key field*. Each value of the foreign key field matches exactly one value of the primary key field in the related table. A foreign key should have the Indexed attribute.

If both the primary and foreign key fields do not have the Indexed attribute when you create the relation, 4D assigns this attribute automatically.

Each value in a foreign key field is equal to one value of the primary key field in another table. In this example, a value of the foreign key field in [Employees] matches exactly one value of the primary key field in [Companies]. The foreign key field is also indexed but its values are non-unique (e.g., several employees may work for the same company).

In some database designs, the values of the primary key field are assigned by the database automatically — either by assigning a sequence number that 4D generates or by a user-written method. Such a procedure guarantees the uniqueness of the key field. For example, if the primary key field in the [Companies] table is a sequence number rather than the company name, it would be possible for users to enter several companies with the same name but different addresses. Also, if a company name changes, the user could make the change to the database without disturbing the relation between the two tables.

If the user is permitted to enter the value of the primary key field, you should use both the Unique and Can't Modify properties to check for uniqueness of the initial entry and to prevent users from subsequently changing the entry to a non-unique value. If you elect not to use the Can't Modify attribute, you will need to take other measures to prevent users from creating "orphaned" records in any related tables by making changes to the values of the primary key field.

When relations are established, you can read and write values in one table while working in the related table. For example, when you enter a company name in an employee's record, 4D searches for that company in the [Companies] table and displays the company address and phone number in that employee's record. When you view a company's record, 4D searches in the [Employees] table for all the employees who work at that company and displays their records in the company record.

These relations can be invoked automatically (i.e., with no programming on your part) or you can choose to use manual relations. In the latter case, you use methods to load and unload related records and control the creation, modification, or deletion of related records. Manual relations are sometimes preferable in complicated structures in which more than two tables are related to one another and you need to control the loading and unloading of related records.

You can choose to use automatic relations by selecting the appropriate properties at the time the relation between the tables is specified. For more information, see the section "Automatic and Manual Relations" on page 302.

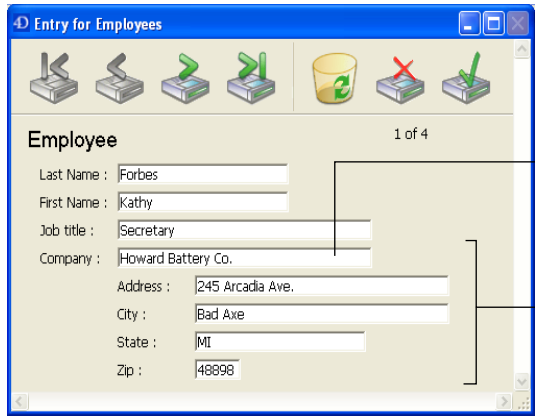
The One Table and the Many Table

When you create a relation between two tables, the table containing the primary key in the relation is called the *One table* and the table containing the foreign key in the relation is called the *Many table*. The tables are called the One table and the Many table because one record in the One table relates to many records in the Many table and many records in the Many table relate to one record in the One table. This type of table relation is called a *Many to One* relation.

In the relation between employees and companies, the [Companies] table is the One table and the [Employees] table is the Many table. One company record relates to several employees (i.e., all the people who work for that company) and several employees relate to one company (i.e., the company for which they work). For instance, there may be one record for Acme in the [Companies] table but many records of people employed by Acme in the [Employees] table.

When any record in the [Employees] table is made current, 4D loads the corresponding single record from the [Companies] table. If any fields have been included from the [Companies] table, the values for these fields are automatically displayed.

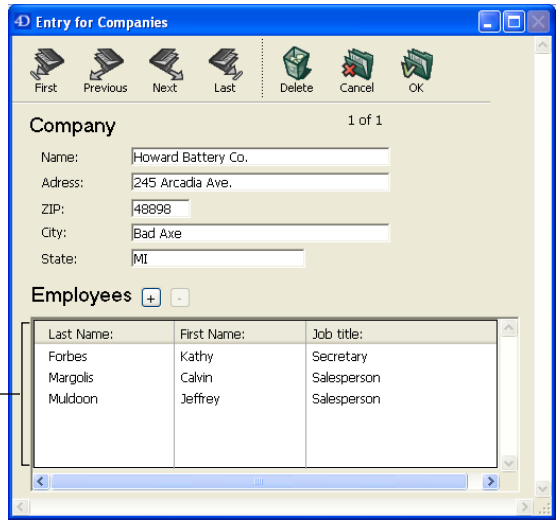
The figure below shows how the company name in an [Employees] table record specifies a record in the [Companies] table so that the [Employees] table record can display the company's address and phone number.



Conversely, when a record in the [Companies] table is made current, 4D creates a selection of records in the [Employees] table and displays them on the form. Since the relation specifies several records in the other table, the names and titles of many employees can be displayed.

Note Only records currently displayed in the form are loaded into memory.

The figure below shows how a company name in a record in the [Companies] table specifies several records in the [Employees] table so that the [Companies] table record can display a list of people employed by that company.

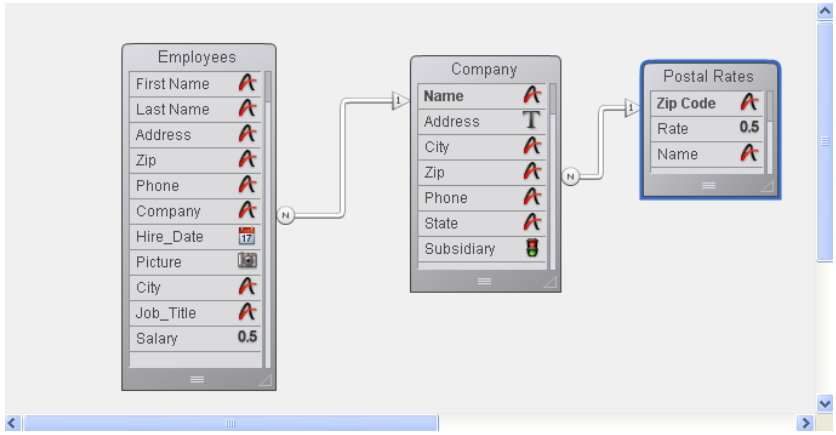


Records from [Employees] table displayed on [Company] form

The distinction between the One table and the Many table is specific to a particular relation. A table may be the One table in one relation and the Many table in another. A table in a relation can only have one primary key, but it can have several foreign keys.

For example, suppose you decide to send a package of sample merchandise to everyone in your [Employees] table. You add a [Postal Rates] table that contains zip codes and the postal rate for each zip code. Using this structure enables you to print an address label for each employee that includes the amount of postage needed to mail the package.

The figure below shows the [Postal Rates] table added to the database structure:



The Zip Code field in the [Postal Rates] table is its primary key, so the [Postal Rates] table is the One table. The Zip field in the [Companies] table is the foreign key field for this relation. Since the Zip field is a foreign key, it can have non-unique values. The Zip field will contain duplicate Zip codes for companies that are near each other. The [Companies] table is therefore the Many table in relation to the [Postal Rates] table.

Whether a table is a One table or a Many table, therefore, depends on its relation to the other table. The [Companies] table is the Many table in relation to the [Postal Rates] table and it is the One table in relation to the [Employees] table.

Automatic and Manual Relations

Relations can be either automatic or manual.

In an automatic relation, whenever a record in a related table is made current, 4D selects the corresponding record or records. The record or records so specified can then be viewed, printed, modified, or used in searches and sorts. No programming is required.

In a manual relation, you dictate whether 4D loads the corresponding record or records into memory. To exercise this control, you use methods. For complete information about creating the methods that control related tables, see the *4D Language Reference* manual.

You would use a manual relation if you wanted to optimize the performance of specific applications that do not need all corresponding records loaded each time. For example, if your structure relates three or more tables together, you may want to control when related records are loaded into memory. You would also use a manual relation if you wanted to relate two tables with two separate relations. Only one automatic relation can exist between two tables. Any number of manual relations can exist between two tables.

Entering Data in Related Tables

You can display fields from the One table in a form for a related table. The user can use these fields to enter and edit the data directly in the records of the related table.

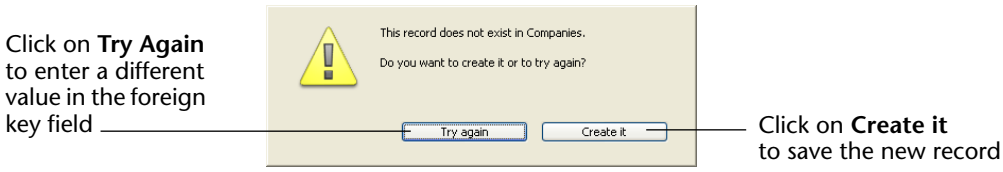
If the relation is automatic, information entered into the related fields is automatically saved in the related field's table.

If the relation is manual, you use the language to display values in related fields and to save the information entered into related tables. You use methods to transfer and save any entered data.

The user enters records in the One table as in any other table — by typing information into an input form or by importing the data. For instance, you might have a subform in the [Companies] table that displays employee data from the [Employees] table. If automatic One to Many relations have been established, any information you add or modify in the subform is automatically updated in the [Employees] table. For more information about subforms, see the "Adding a Subform to the Form" paragraph on page 621.

Occasionally, a user needs to create a new record for the One table while creating a record in the Many table. For example, suppose that, while creating a record in the [Employees] table the user enters a company name that doesn't exist in the [Companies] table. If automatic Relate One relations have been established, 4D automatically gives the user the opportunity to create a new record in the [Companies] table.

If the **Prompt if related one does not exist** property is selected, 4D displays the following message when a user enters a company name that does not exist in the [Companies] table.



4D requires that the corresponding record exist in the related table. The chance to create a new record in the One table is automatically provided to the user.

Creating and Deleting Relations

You can create or remove relations via the Structure editor or using the SQL commands of 4D. This section covers the manual creation of relations in the Design environment. For more information about using SQL statements in 4D, please refer to the *4D SQL Reference* manual.

Creating a Relation

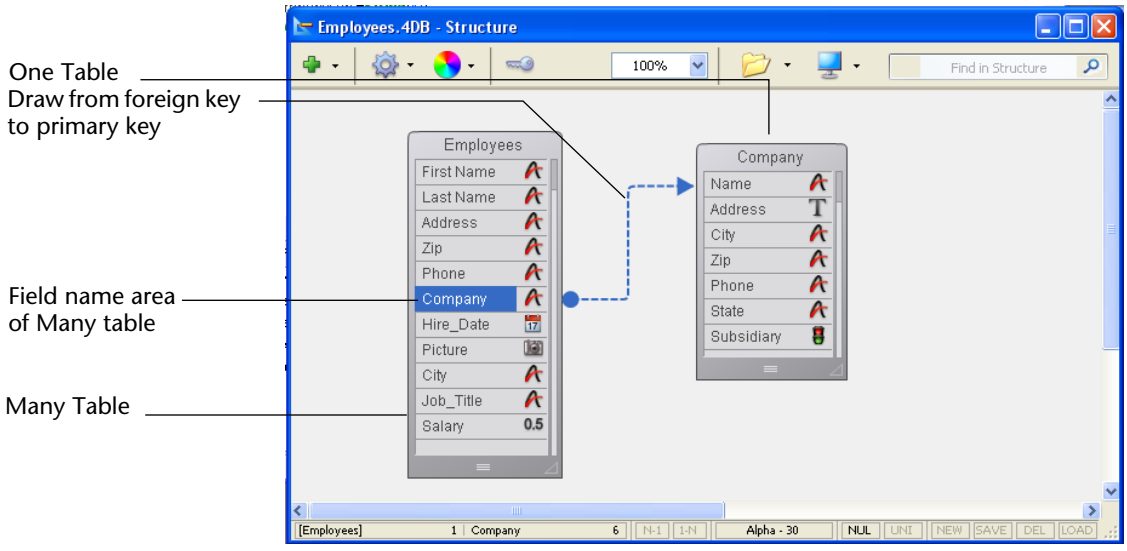
You can create a relation between two tables by manually tracing a line between two fields or using the add objects menu of the tool bar.

► To create a relation using the Structure editor window:

1 **In the Structure editor window, move the pointer over the foreign key field for this relation.**

2 **Hold down the mouse button and drag toward the table to be related.**

As you move the pointer, 4D selects the field and draws the relation, as shown below.



3 Drag to the primary key field in the One table and release the mouse button.

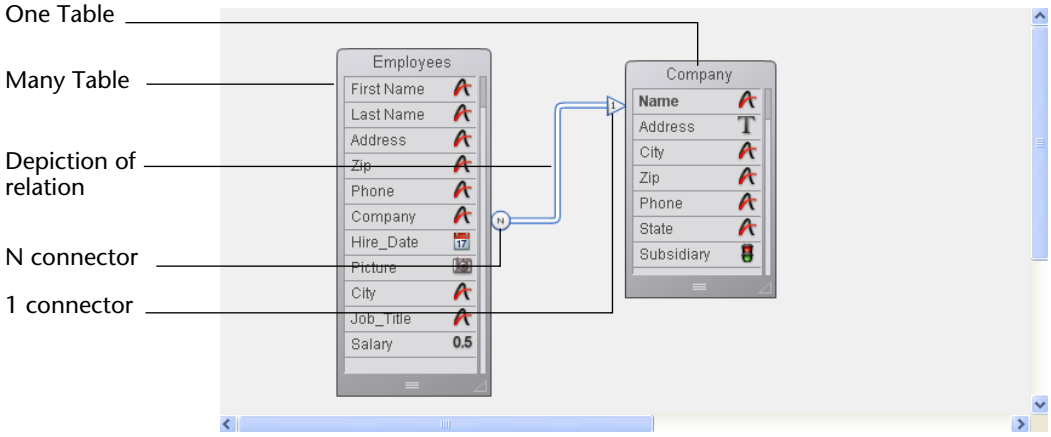
► To create a relation using the add objects menu:

1 Select two fields of the same type belonging to two different tables.

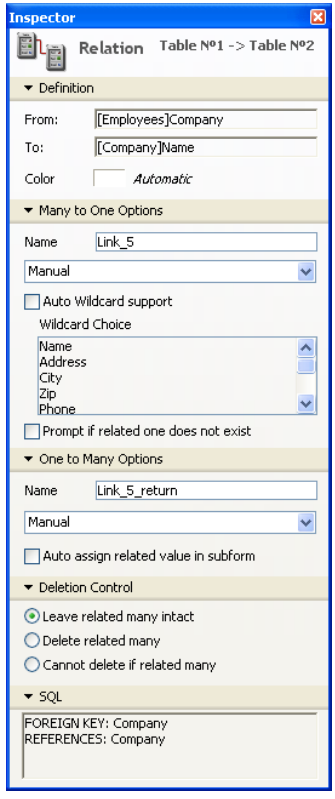
The order of selection will determine the direction of the relation. The first field selected is considered as the foreign key (Many field) and the second as the primary key (One field).

2 In the add objects menu of the Structure editor tool bar, choose the Relation option:

4D displays the relation as an arrow between the two tables in the Structure editor. The shape of the beginning and end connectors as well as the N and 1 characters shown within them indicate the direction of the relation:



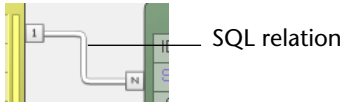
The Inspector palette displays the properties of the relation.



For more information about configuring relation properties, please refer to the section “Setting Relation Properties” on page 308.

Relations Created via SQL

The appearance of relations is different when they have been generated via SQL (square shaped connectors):



Deleting a Relation

4D lets you delete a relation at any time.

▶ To delete a relation:

1 **Select the relation to be deleted.**

When a relation is selected, its outline turns blue.

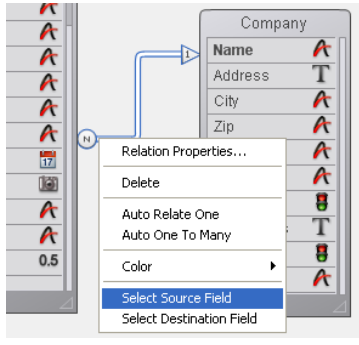
2 **Right-click on the relation and choose the Delete command from the contextual menu.**
OR
Press Delete or Backspace.

In both cases, a warning dialog box appears so that you can confirm or cancel the operation. If you confirm it, 4D removes the linking arrows and the tables are no longer related. Any indexes of Many and One fields are not deleted.

Reestablishing a Relation

You can reestablish any relation at any time. You would do so, for example, if you mistakenly draw the relation between the wrong fields or if you want to change a relation property. 4D lets you reestablish a relation simply by drawing the relation line again.

In the case of complex structures, you can select fields in the source and destination tables of a relation using the **Select Source Field** and **Select Destination Field** commands, located in the contextual menu of the relation.



To reestablish a relation using the same two fields, double-click on the relation in the Structure editor. 4D displays the relation properties in the Inspector palette so that you can make any necessary changes.

To reestablish a relation using a different field in the One table, redraw the relation beginning from the Many field.

To reestablish a relation using a different field in the Many table, first remove the faulty relation and then draw the correct relation line again.

Setting Relation Properties

You must have at least two tables in your database to create a relation. You create a relation by drawing a line between two fields.

The field where you start drawing must be a foreign key field in the Many table and the field where you end must be the primary key field in the One table. Using the Companies database example in this chapter, you would start drawing in the [Employees] table and end in the [Companies] table. Remember:

You always draw a relation from the Many table to the One table.

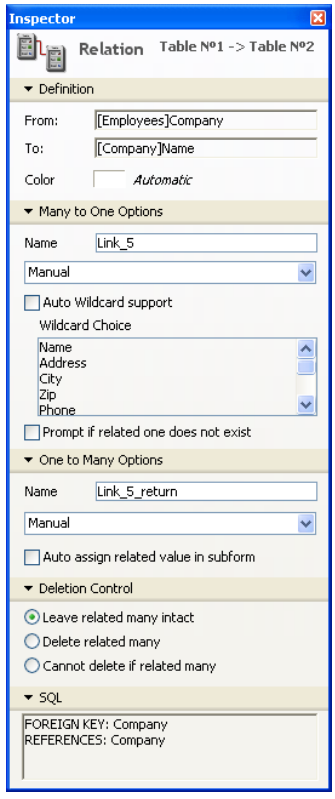
The related fields must have the same field type. For example, the Company field in the [Employees] table and the Name field in the [Companies] table can be related because they are both Alpha fields.

You can use these field types for the primary and foreign key fields:

- String (Alpha and Text)
- Number (Real, Integer, Long Integer, Integer 64 bits or Float)
- Time
- Boolean
- Date.

Primary key fields are usually indexed (not mandatory). If this is not already the case, 4D automatically indexes them.

The relation properties can be configured using the Inspector palette or, for certain ones, using the contextual menu that appears when you right-click on the relation:



Inspector

Contextual menu

Note For more information about using the Inspector, please refer to the "Inspector Palette" paragraph on page 246.

Definition

The Definition area identifies the foreign and primary key fields.

From

The "**From**" field is the foreign key of the Many table for this relation. In the contextual menu, this field is called the "source field."

To

The "**To**" field is the primary key field of the One table. In the contextual menu, this field is called the "destination field."

You draw the relation line *from* the foreign key field in the Many table *to* the primary key field in the One table.

Color This pop up menu lets you set the color of a relation as it will be displayed in the Structure editor. The color of a relation is not linked to its properties.

Note The color of a relation can also be set using the contextual menu for relations.

Many to One Options

The Many to One options affect what happens when a record from the Many table is opened.

Name Entry area for naming Many-to-One relation (optional).

Manual/Automatic This menu can be used to set the operating mode of the relation linking the Many table to the One table. If you select the **Automatic** option, automatic functions will be enabled. For example, when a record from the [Employees] table is open in the Application environment, the corresponding company record is selected in the [Companies] table. This allows 4D to display information about the company the where the employee works if you want.

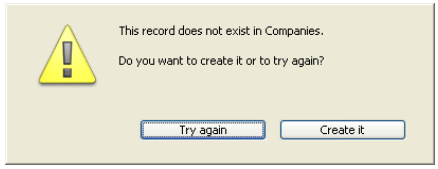
By default, the **Manual** option is selected. In this case, you must manage the loading and unloading of the related record of the One table using language routines.

For more information about this, please refer to the "Automatic and Manual Relations" paragraph on page 302.

Note The **Auto Relate One** property can also be set using the contextual menu for relations.

- **Auto Wildcard support**: This check box has the effect of invisibly appending the wildcard character (@) to any value entered in the foreign key field of the Many table when the user tabs or clicks out of the field. If the user enters a partial value, 4D looks for a matching value in the related One table. If 4D finds only one match, it automatically completes the entry. If 4D finds more than one possible match, the user is presented with a list of values from which to choose. For a complete description of this process, see the section "Entering Data in Related Tables" on page 303.

- **Prompt if related one does not exist**: This check box forces 4D to display a dialog box that lets a user create the related One record if it does not exist. By default, when you enter a value in a related field from the Many table, 4D checks to see whether a matching record already exists in the related One table.
If 4D cannot find a match, the following dialog box is displayed:



This dialog box allows the user to create a corresponding record in the One table while you are entering a record in the Many table. For instance, suppose that you have an Invoicing database that contains an [Invoices] table and a [Customers] table. If you enter an invoice in the [Invoices] table and the customer to whom the invoice belongs does not already have a record in the [Customers] table, 4D will ask you if you want to create the corresponding record in the [Customers] table when you validate the record in the [Invoices] table.

You can eliminate this dialog box by unchecking the **Prompt if related one does not exist** check box. Removing this dialog box is useful when you need to manage the creation of the related One record using a method.

Entering Data in Related Tables

You can display fields from the One table in a form for a related table. The user can use these fields to enter and edit the data directly in the records of the related table.

If the relation is automatic, information entered into the related fields is automatically saved in the related field's table.

If the relation is manual, you use the language to display values in related fields and to save the information entered into related tables. You use methods to transfer and save any entered data.

The user enters records in the One table as in any other table — by typing information into an input form or by importing the data. For instance, you might have a subform in the [Companies] table that displays employee data from the [Employees] table. If automatic One to Many relations have been established, any information you add or modify in the subform is automatically updated in the [Employees] table. For more information about subforms, see the "Adding a Subform to the Form" paragraph on page 621.

Occasionally, a user needs to create a new record for the One table while creating a record in the Many table. For example, suppose that, while creating a record in the [Employees] table the user enters a company name that doesn't exist in the [Companies] table. If automatic Relate One relations have been established, 4D automatically gives the user the opportunity to create a new record in the [Companies] table.

If the **Prompt if related one does not exist** property is selected, 4D displays the following message when a user enters a company name that does not exist in the [Companies] table.

Click on **Try Again** to enter a different value in the foreign key field

Click on **Create it** to save the new record

4D requires that the corresponding record exist in the related table. The chance to create a new record in the One table is automatically provided to the user.

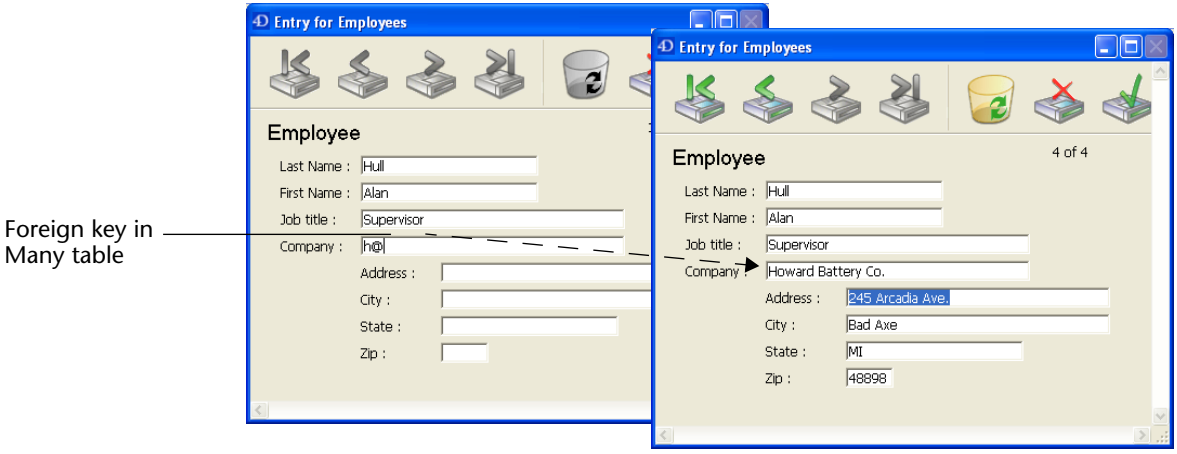
Wildcard Choice List

The list of wildcard fields lets you select an additional field that will be displayed in the list of values (that appears when the user enters the wildcard @ in the related field during data entry). Normally, it is preferable to select the field that best identifies the record.
Here's how this works during data entry: 4D allows the user to look up values in the One table when entering data into the foreign key field in the Many table. The user simply uses the standard wildcard character (@) in the related field. Doing so causes 4D to search for the corresponding entry in the related One table.

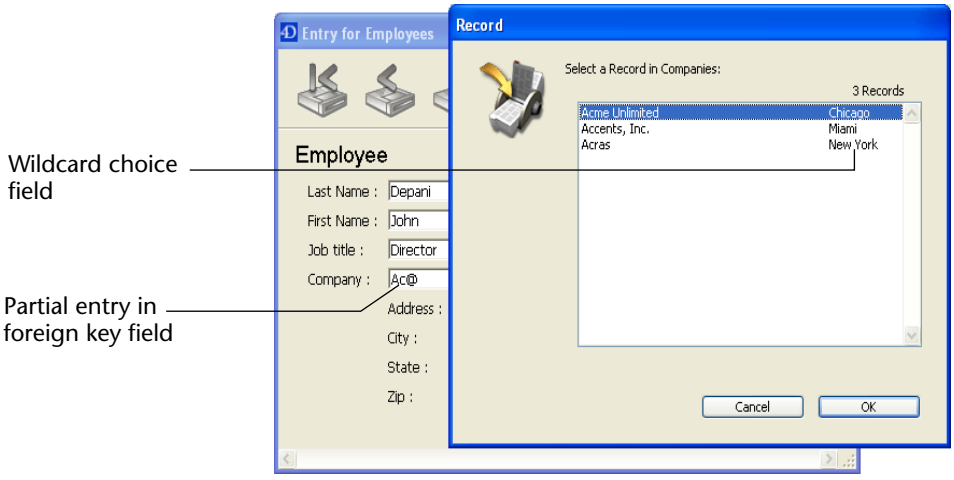
The wildcard character can be used in two ways: to complete a partial entry or to display a list of valid entries. When a list is displayed, the user can select the entry from the list. An additional field, the wildcard choice field, can be displayed with the related field.

For example, suppose the user is creating a record in the [Employees] table. Instead of typing Acme Unlimited in the Company field, the user can type Ac@ and then press Tab to move to the next field. Because @ is the 4D wildcard character, this entry means “this value starts with “Ac” and is followed by anything else.” 4D looks in the related table for the record which matches this entry. If it finds one, it completes the entry and selects the next field in the data entry order.

The figure below shows how this use of the wildcard works.



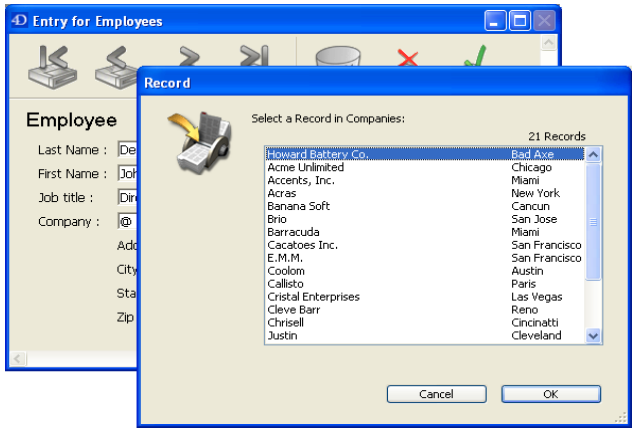
If 4D finds more than one entry that meets the requirement, it displays a list of entries so that the user can select the correct one. The figure below shows such a list being displayed:



You can specify a second field for the list to help the user decide which company to select. The second field is the wildcard choice field you selected in the Inspector palette when you created the relation.

The figure above shows the list of companies displaying the city as well as the company name. This wildcard choice field assists the user who doesn't know whether the company is named Accents, Inc. or Acme Unlimited, but remembers that the company is located in Chicago.

To see a list of all companies in the [Company] table, the user enters @ only. 4D then displays a list of all the companies so that the user can select the correct one. The figure below shows a complete list of companies being displayed.



Note The record selection window can be resized.

To see the list of all the companies in the [Company] table, you can enter only the @ character. 4D then displays the complete list of all the companies of the table so that the user can select the correct one.

One to Many Options

The One to Many options control automatic relations in the other direction.

Name

Entry area for naming One-to-Many relation (optional).

Manual/Automatic

This menu can be used to set the operating mode of the relation linking the One table to the Many table. If you select the **Automatic** option, automatic functions will be enabled. For example, when a record from the [Company] table is open in the Application environment, the related records of the [Employees] table are loaded. This allows 4D to display the employee records of this company in an included subform.

By default, the **Manual** option is selected (no automatic functions).

For more information about this, please refer to the "Automatic and Manual Relations" paragraph on page 302.

Note The **Auto One To Many** property can also be set using the contextual menu for relations.

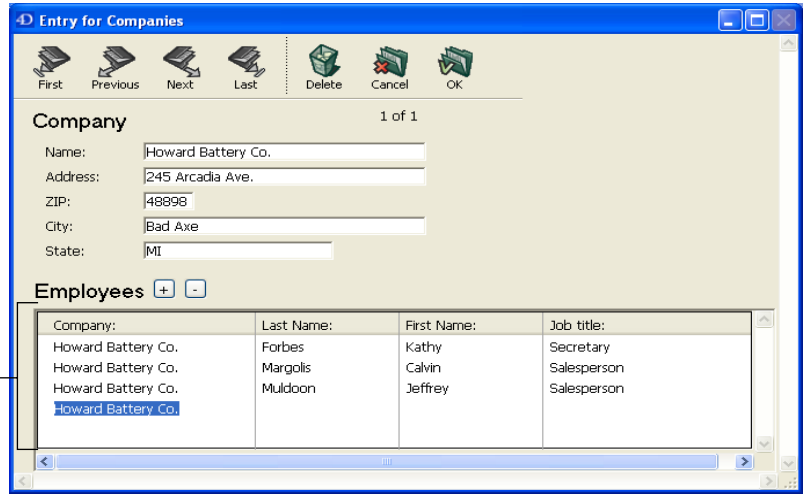
Auto Assign Related Value in Subform

This option is used to automatically assign the value of the primary key field in the One table to the foreign key field in the Many table during data entry. This option is only useful when the **Automatic** option has been selected for the return relation.

This option affects data entry when an input form in a One table has a subform of a related Many table (for information on subforms, see the section "Adding a Subform to the Form" on page 621). If **Auto assign related value in subform** is selected, a user can add records to the subform (i.e., the related Many table) and have the related value automatically assigned to the fields of the related table.

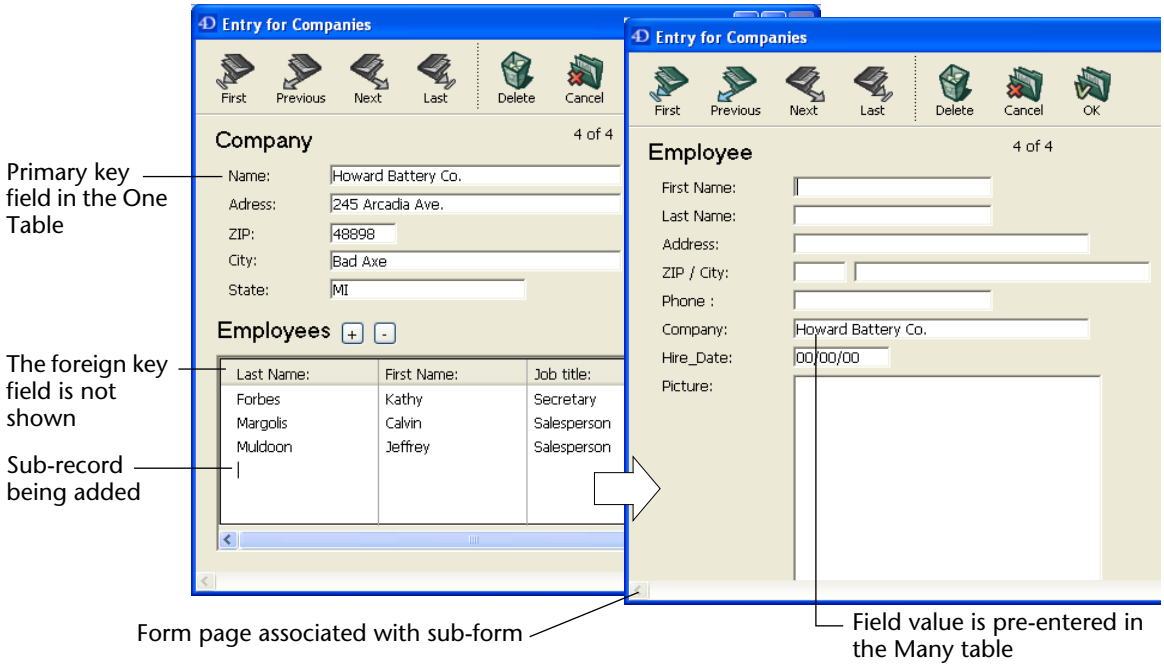
In the relation between the [Company] table and the [Employees] table, the [Company] table is the One table and the [Employees] table is a related Many table. Each company has one record in the [Company] table and several records in the [Employees] table.

When the foreign key field of the Many table is displayed in the sub-form, you can see the effect of the option: the field value is automatically copied into the sub-form each time a sub-record is added.



This option also works when the foreign key field of the Many table is not shown in the sub-form: the primary key field value is automatically copied “internally” into the foreign key field.

You can make sure by passing to page mode:



When the **Auto assign related value in subform** option is not checked, the sub-records created are not automatically related to the appropriate record of the Many table.

In this case, you must link the sub-record:

- Either manually, if the key field is displayed: in the previous example, you simply need to enter “Howard Battery Co.” manually in the Company name field of each sub-record,
- Or by programming: you just need to execute a line of code of the [TableN]Field1:=[Table1]Field1 type when creating the sub-record.

Note If you modify the related field value in the One table after you have created records in the Many table, **Auto assign related value in subform** has no effect and you must either manually assign the related field value or use the language.

Deletion Control

The Deletion Control options regulate record deletion in the Many table when a record is deleted in the One table. Normally, the user cannot delete records in a table unless it is the current table. This means, for example, that to delete records from the [Employees] table, it must first be made the current table. You make a table the current table by choosing it in the List of tables dialog box in the Design environment.

The following Deletion Control options can be set only if the Allow Deletion Control check box in the Preferences dialog box is checked. For more information, see the section "Data Management Page" on page 193.

Leave Related Many Intact

Selecting this radio button allows the user to delete a record in the One table, leaving the corresponding records in the Many table intact. This leaves records in the Many table without any corresponding related record in the One table. The only effect is to render the information from the One table unavailable. No record from the One table is loaded when a record corresponding to the deleted record is loaded in the Many table.

Delete Related Many

Selecting this radio button instructs 4D to automatically delete all related records in the Many table when the user deletes a record in the One table. This property ensures that no related Many records become "orphaned" when the corresponding related One record is deleted.

Cannot Delete if Related Many

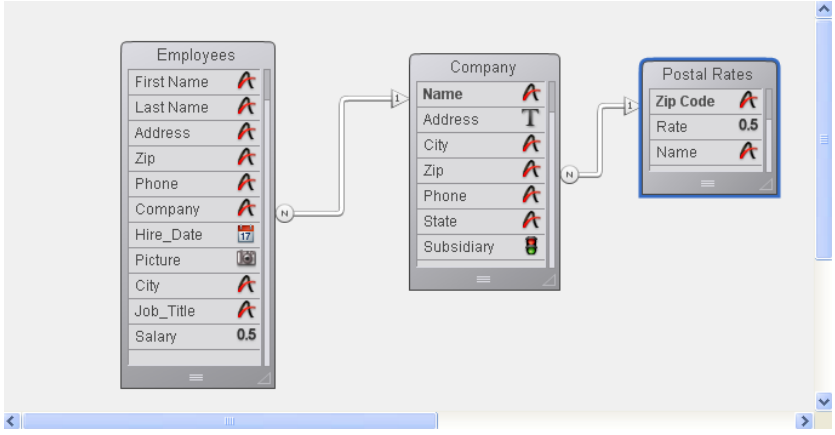
Selecting this radio button instructs 4D to prohibit the user from deleting a record in the One table if there are related records in the Many table. This property ensures that no records are mistakenly deleted.

Note that you can freely delete records from the Many table, no matter which choice is made.

The **Delete related many** and **Cannot delete if related many** radio buttons enforce what is called *referential integrity* in database theory. When referential integrity is in effect, 4D ensures that each record in a related Many table will always be associated with exactly one record in the related One table.

If you set the Deletion Control option to either **Delete related many** or **Cannot delete if related many**, 4D automatically adds the Indexed, Can't Modify, and Unique attributes to the primary key field in the One table. You cannot remove these attributes unless you first change the Deletion Control setting to **Leave related many intact**.

If you have several related tables, deletion control is activated for each relation as in a chain. For instance, suppose you have the structure shown below. If a Zip code is deleted from the [Postal Rates] table (a One table) and **Delete related many** has been selected for each relation, 4D first deletes the records for the corresponding companies in the [Company] table and then deletes the records of all the employees who work for those companies in the [Employees] table.



When confronted with contradictory Deletion Control settings, 4D does not allow the deletion to occur. For instance, if **Delete related many** is selected for the relation between the [Company] table and the [Postal Rates] table but **Cannot delete if related many** is selected for the relation between the [Employees] table and the [Company] table, no deletion will occur and the records in the [Company] and [Employees] tables will remain intact.

SQL

The SQL area of the Inspector palette provides information that is useful for working with the structure via the SQL language.

For relations, the area indicates FOREIGN KEY and REFERENCES properties.

Other Relation Types

Until now, we have been discussing the process of creating the most common type of relation — a relation between a One table and a Many table — called a Many to One relation. However, you can also create One to One and Many to Many relations. These types of relations are described in this section.

One to One Relations

One to One relations are used only in special cases since tables that are related on a one-to-one basis could be combined into a single table. Here are some reasons to use a one-to-one relation:

- You have large Text, Picture or BLOB fields in the database. These fields would slow down the database if they were loaded into memory when a record is made current. By placing text, pictures and BLOBs in another table, you can load the data only when needed.
- You have a very large number of fields and need to divide them into logical groups. Separate tables can make the database faster and easier to use.
- You want to limit access to certain fields. If you use separate tables, you can assign different access privileges to each table.

Many to Many Relations

Sometimes you need to relate many records in one table to many records in another table. This is called a Many to Many relation.

An example of a Many to Many relation is a database that tracks class enrollment. Suppose that this database has two tables, [Students] and [Classes]. A student may enroll in many classes and a class may have many students. You want to see all the classes that a student has enrolled in and you want to see all the students enrolled in each class.

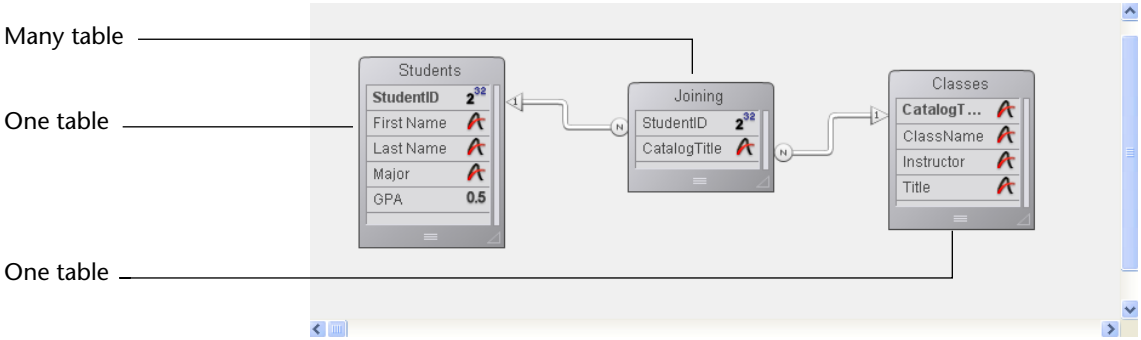
Other examples of Many to Many relations include the following.

- **[Suppliers] and [Products]**: Each supplier provides many products and each product may be provided by several suppliers.
- **[Employees] and [Account]**: Each employee works on many accounts and each account may be worked on by several employees.
- **[Movies] and [Actors]**: Each movie involves several actors and each actor may appear in several movies.

You can use 4D to create automatic Many to Many relations. The key is to create an intermediate table which is related to the other tables using Many to One relations. You can then create input and output forms that handle all the necessary record tracking and data display.

This section describes how to use automatic relations to handle a Many to Many relation.

The figure below shows the enrollment database with three tables, [Students], [Classes], and [Joining]. This database structure is used throughout this section to explain how an automatic Many to Many relation works.



The [Students] table is a One table. It contains one record for each student, including their name, major, and GPA. The Student ID field identifies each student uniquely.

The [Classes] table is also a One table. It contains one record for each class, including the class name and the instructor. The CatalogTitle field identifies each class uniquely.

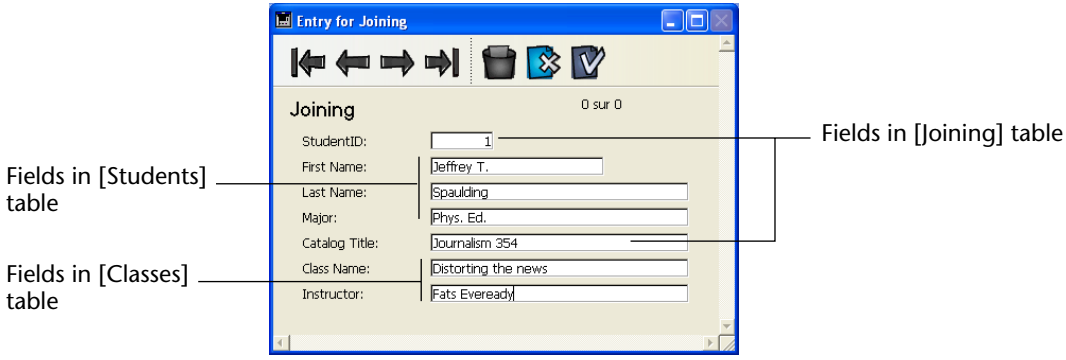
An intermediate table, [Joining], is the Many table for both of the other tables. It contains records for many students and many classes. Forms for this table are used for entering data into both of the other tables, and for displaying information in each of the other tables.

The use of three tables ensures that the data is stored efficiently. A student's complete record is stored only once. Each class has one record, stored only once. Records that relate students to classes are stored once for each enrollment. All of the information, however, is available in any combination.

Entering Data with Many to Many Relations

You use the intermediate table — in this example the [Joining] table — to enter and display information from both of the other tables. Each record that you enter in the [Joining] table is related to both of the other tables (a student and a class). The records from the [Joining] table contain only the two pieces of information that establish the relation: the student ID and the catalog title.

Here is an example of a new record being entered in the [Joining] table:

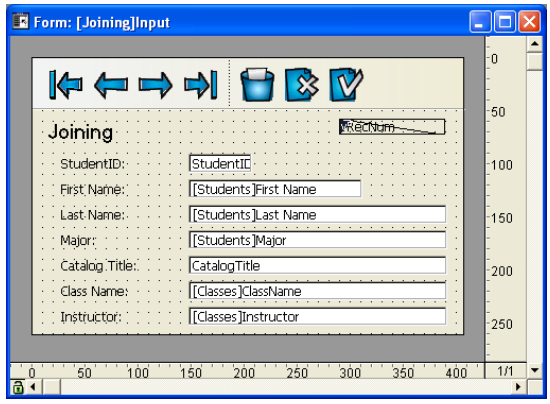


This record defines Jeffrey T. Spaulding as enrolled in a Journalism class. This record actually combines information from the other two tables.

A similar record exists for each class in which the student is enrolled. Only the Student ID and Catalog Title fields are actually stored in the [Joining] table. Each record catalogs a particular student taking a particular class.

Note When a record in the [Joining] table is loaded (as when creating such a record), it automatically creates a selection of records in the related tables. The selection consists of the corresponding student and class records. If you switch to either of the other tables, only a single record is displayed. To display all the records, choose **Show All** from the **Queries** menu.

The input form for this record is shown below. Notice that it contains fields from both the [Students] and [Classes] tables.

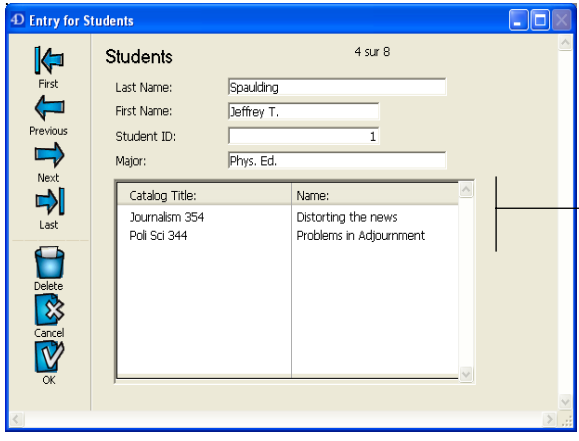


Data is entered only in the Student ID and Catalog Title fields. When a student ID is entered, 4D finds the student information in the related Students table and displays it in the Last Name, First Name, and Major fields. Likewise, when a Catalog Title is entered, 4D finds class information in the [Classes] table and displays it on the input form.

Displaying Information in a Subform

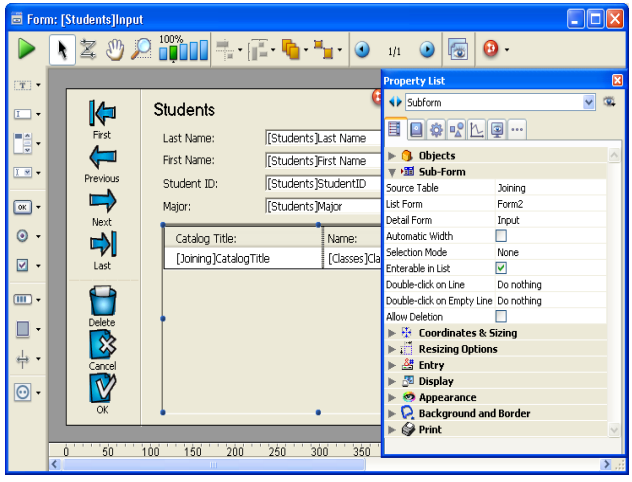
You can display information from these three tables using subforms. In the student's record, you can display all the classes in which he or she is enrolled. In the class record, you can display all the students enrolled in a particular class.

To display classes in a student's record, you use a subform. For information about creating subforms, see the section "Adding a Subform to the Form" on page 621.



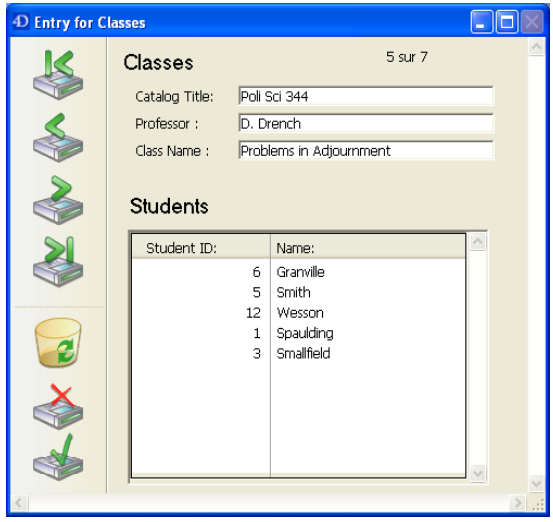
The record shown above is in the [Students] table. It shows information about the student at the top of the record. The information about the two classes that he is enrolled in is drawn from the [Joining] table where the enrollment information is kept.

Here is the form for this record as well as the Property List with the subform properties selected:



Notice that the subform is for the [Joining] table, not the [Classes] table. The [Joining] table contains the records that relate the student's record to the class records. The subform contains the ClassName field from the [Classes] table. Because of the relation between the [Joining] and [Classes] tables, 4D can display the correct class name automatically.

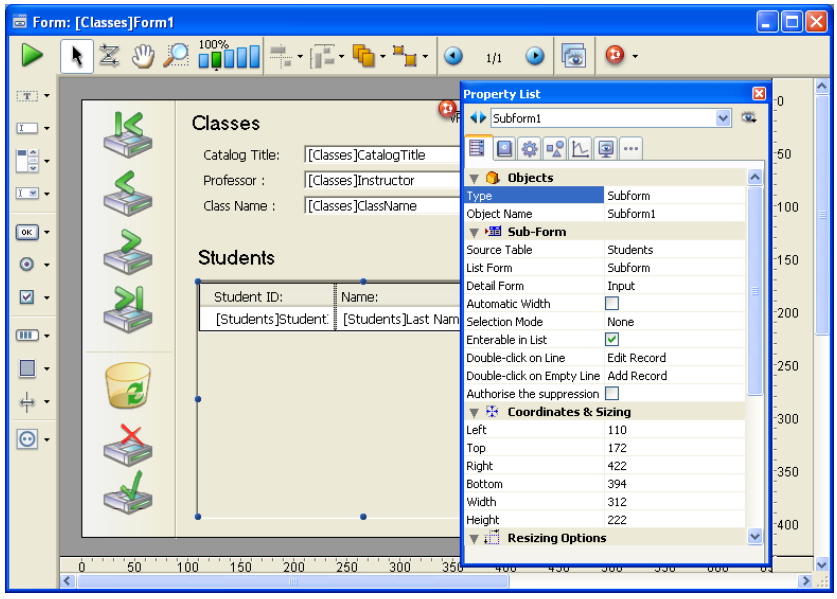
Here is a record that shows the students who are enrolled in a class:



This is a record from the [Classes] table. It shows class information and lists the students enrolled in the class.

The information about the students is also drawn from the [Joining] table since that table contains the records that relate the classes to the students enrolled in them.

Here is the form for the record shown above:



In the above examples of subforms, you can enter records in any of the fields shown.

For example, to enter a new student into a class record, you simply tab to the last student record shown in the subform and press **Ctrl+:** (Windows) or **Command+:** (Mac OS)[1] to create a new record. When you enter the appropriate catalog title, the remainder of the information is entered in the record. For information about entering and deleting records in subforms, refer to the "Subforms Options" paragraph on page 629.

1. This default shortcut can be modified in the application Preferences (see the section "Shortcuts Page" on page 171).

Creating Reports

Quick reports that include information from both the [Classes] and [Students] tables are typically generated from the intermediate table. If you create form reports for printing, you can use subforms to print information from either of the related One tables.

Here is a simple Quick report that lists all the classes and the students that are enrolled in each class.

Class Name	First Name	Last Name
Auto Mech 101	Rufus T.	Firefly
Ceramics 211	Otis B.	Driftwood
	Jeffrey T.	Spaulding
Journalism 354	Jeffrey T.	Spaulding
Pol Sci 344	Jeffrey T.	Spaulding
	Otis B.	Driftwood
	Rufus T.	Firefly

For complete information about creating and printing Quick reports, refer to the chapter “Quick Reports” on page 699. For complete information about using forms for printing reports, refer to the chapter “Output Displays and Reports” on page 635.

Analyzing Database Relations

The relations that you establish in a database play an important role in the operation of the database by controlling the flow of information between the tables.

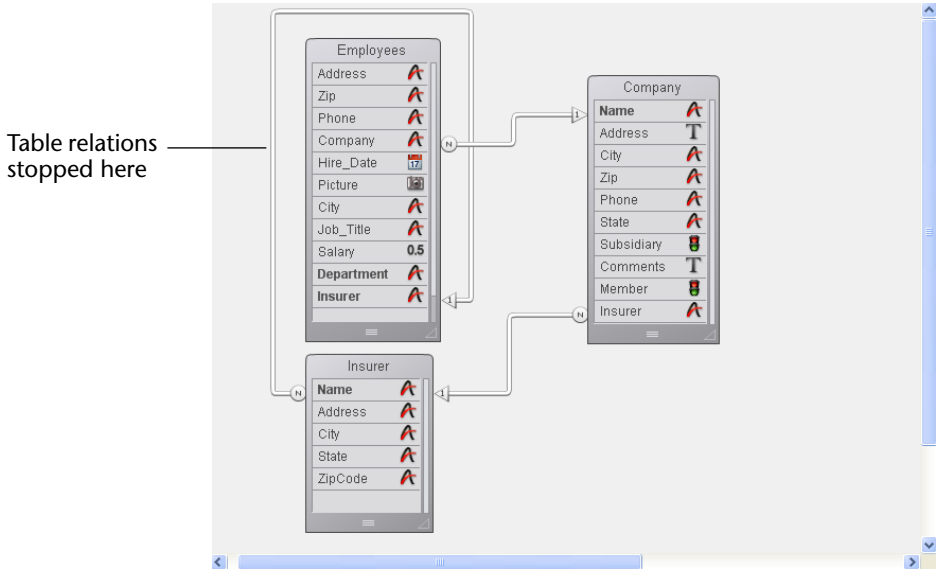
If a record with an automatic relation is loaded from disk using an input form, the corresponding record or records from the related table are selected. If a relation selects only one record in a related table, that record is loaded from disk. If a relation selects more than one record in a related table, a new current selection of records is created for that table and the first record in the current selection is loaded from disk. The record that is loaded from disk is called the *current record* for the table.

In the examples in this chapter, relations have been established between no more than three tables. In the real world, relations are often created between several tables and are activated one after the other, as in a chain. Each time a relation is activated, 4D creates a selection of records in the related table and loads a record from disk. The record that is loaded from disk becomes the current record for the table and — if the table has an automatic relation —4D creates a selection and loads a current record in the next related table in the chain, and so on.

If the table relations have not been set up properly, the circulation of information between tables can become disorderly or corrupt. The following cases alert you to relational structures of which you should be aware.

Circular Relations

A circular relation is one in which table relations are set up so that the transfer of information will loop indefinitely. The figure below shows a circular relation in which the [Employees] table relates to the [Company] table, which relates to the [Insurer] table, which relates back to the [Employee] table.



When a record in the [Employee] table is loaded from disk, 4D loads the related company record from the [Company] table. This becomes the current record for the [Company] table, which in turn loads the related insurer record from the [Insurer] table.

If the table relations were allowed to continue, the records related to this insurer (all the people insured by the company) would be selected in the [Employee] table and the first record in that selection would be the current record. Note that this current record may be different from the current record that started this progression. In this situation, 4D has no way of knowing which record is really the current record.

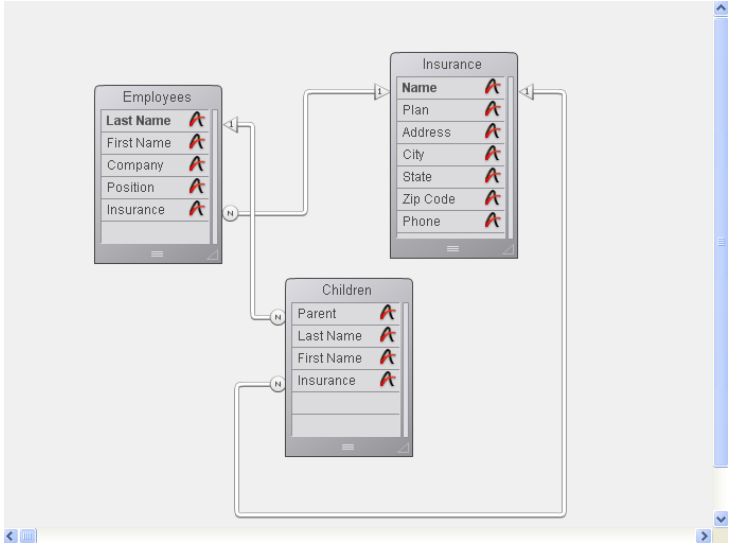
When 4D encounters this kind of circular relation, table relations are stopped at the last table in the chain. In this case, the relation between the [Insurer] table and the [Employee] table is not carried out.

Multiple Links to the Same Table

A similar conflict between current records occurs if you have more than one link to the same table.

Since you cannot have more than one current record at a time, you cannot manage an automatic table relation in which two or more tables are related to the same table.

The following illustration shows a database structure in which a table and its subtable both relate to the same table.



When a user is working with a record in the [Employees] table, the related record is loaded in the [Insurance] table and it is made the current record for that table.

However, there is also a relation between the [Children] table and the [Insurance] table. This means that another related record is loaded in the [Insurance] table based on the current record (the first record) in the [Children] table. If the child's insurance company is different from the parent's, this relational structure will cause problems.

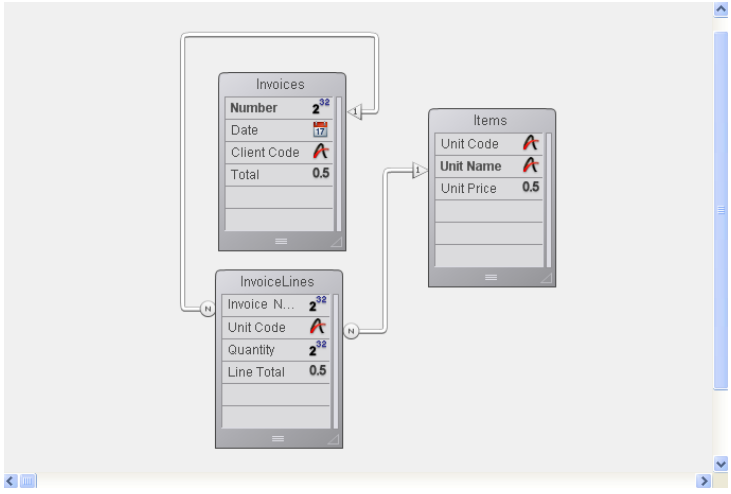
In this case, 4D does not stop the relations from proceeding. Both the relations are carried out, but not at the same time.

If you want to use this kind of structure, you must use manual table relations and control the relations using the commands described in the *4D Language Reference* manual.

Another example of a relational structure that cannot be managed by automatic relations is a structure in which one table has more than one relation to another table. Each time a user modifies either of the related fields in one table, the current record in the other table may change. In this situation, you cannot tell which relation is being activated.

Relations from Multiple Records

Since there is only one current record in a table, relations are not established for all of the records in a selection. Let's take the case of the the Invoices database shown in the structure below.



When a record in the [Invoices] table is being used, a selection of records is created in the [InvoiceLines] table that contains all of the lines for that invoice. But the corresponding record in the [Items] table is selected only for the first item in the [InvoiceLines] table. The selection in the [Items] table does not include information about all the items in the invoice, only the first item.

However, if you place [InvoiceLines] in a subform in the [Invoices] table, 4D calls each invoice line, one at a time, and activates the relationship for each one of them.

Exporting and Importing Structure Definitions

4D lets you export the database structure definition as an XML or HTML file. Conversely, it is possible to use a structure definition saved in XML format to generate a new 4D database on the fly.

These new possibilities meet different needs:

- Allowing structures to be represented in custom formats (reports, tables, etc.) or to be analyzed in other environments,
- Allowing databases to be generated from definition files.

Format of a 4D Structure Definition

4D structure definitions are based on the XML format. You can display a structure definition using a simple text editor. The XML format also allows any type of use to be foreseen, in particular via XSL transformations. Moreover, 4D uses an .XSL file to export the structure definition in HTML format.

A structure definition includes tables, fields, indexes and relations, along with their attributes and the various characteristics that are necessary for a complete definition of the structure. The internal "grammar" of 4D structure definitions is documented through DTD files — also used for the validation of XML files.
The DTD files used by 4D are grouped together in the **DTD** folder, located next to the 4D application. The **base_core.dtd** and **common.dtd** files are used for the definition of the structure. For more information about 4D structure definitions, feel free to consult these files, as well as the comments they contain.

Exporting a Structure Definition

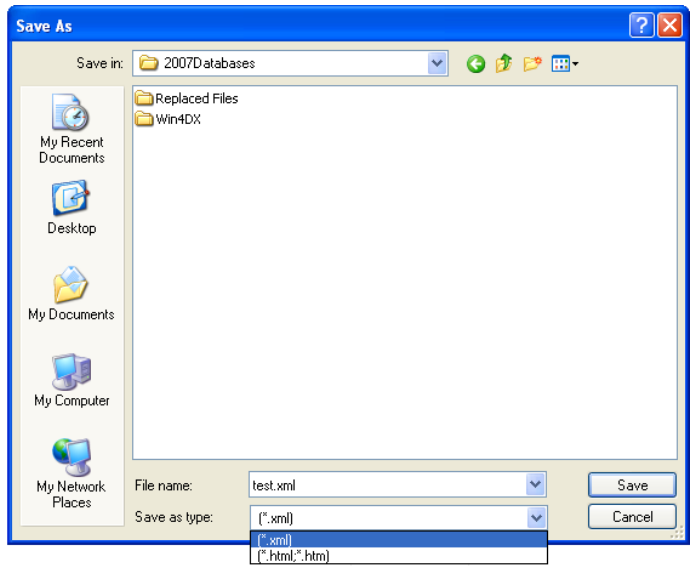
4D lets you export a structure in either XML or HTML format. Choose the format that best meets your needs:

- **XML Format**: An XML-format structure may be viewed using a simple text editor or used in various ways (custom XSL transformation, importing and analysis in another software, etc.).
Choose this format when you want to use the structure definition to create new databases.
- **HTML Format**: This format allows the representation of the structure in the form of a report that can be viewed and printed via a browser.

▶ To export a structure definition in XML:

1 Select the Export > Structure definition to XML file... command in the File menu of 4D.

A standard Save as dialog box appears so that you can specify the name and location as well as the type of file to be exported.



2 Specify the name and location of the export then validate the dialog box.

► To export a structure definition in HTML:

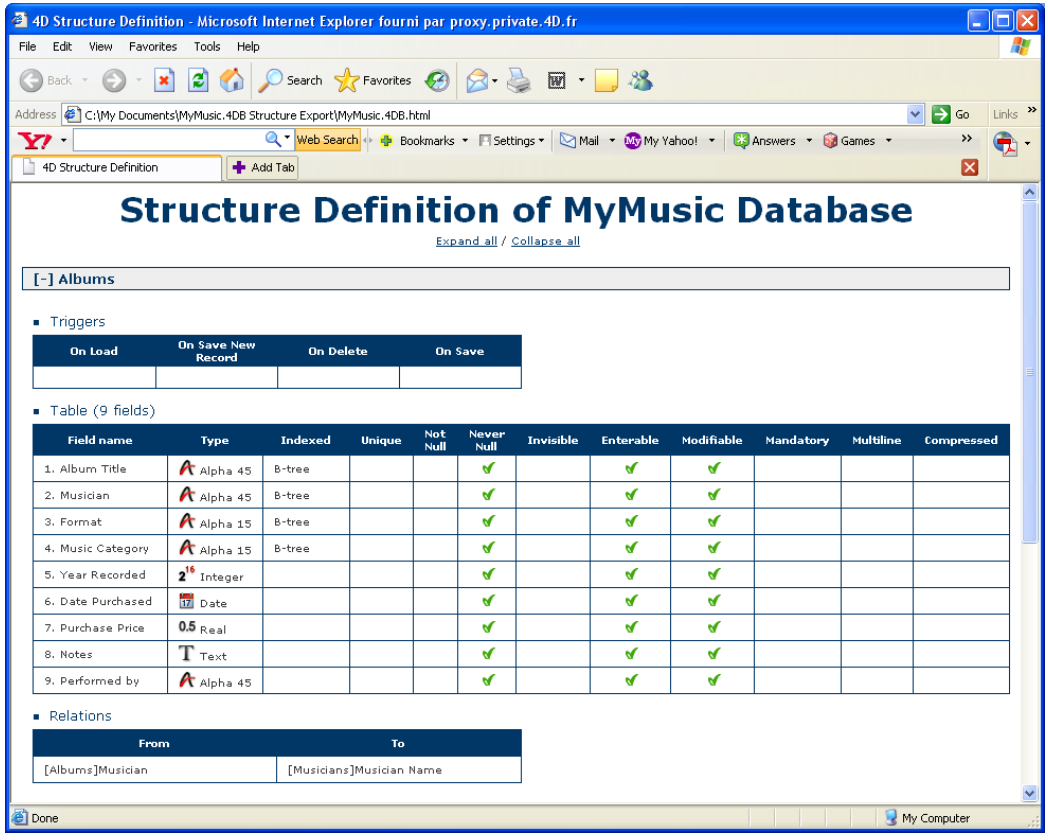
1 Select the Export > Structure definition to HTML file... command in the File menu of 4D.

A dialog box used for selecting a folder appears so that you can designate the location where the HTML files will be stored.

2 Click on Make New Folder or specify an existing folder.

4D automatically creates, at the specified location, a folder named "*Mystructure.4db* Structure Export" that contains the exported items (*Mystructure.4db* represents the name of the database structure file.

A dialog box lets you view the results of the export directly in the default browser. Structure definitions in HTML format look like this:



Customizing the XSL Transformation

To generate the HTML pages of the structure definition, 4D carries out XSL transformations by default using the “Structure_to_html.xsl” file placed in the /Resource/*language*.lproj subfolder of the application.

Note If this file is not present, exporting in HTML is not available in the export dialog box.

You can customize these transformations as desired using a custom XSL style sheet file. To do this, simply create a file named “Structure_to_html.xsl” file (you can duplicate the default file) and place it at the same level as the .4db file. 4D will then use this file to generate the structure definition in HTML format.

Creating a Database from a Structure Definition

Structure definitions that are exported in XML format can be used to create new identical databases on the fly. In this case, the structure definition can be considered as a structure template, which it is possible to duplicate at leisure.

An XML structure definition can be used as such or can be modified beforehand via an XML editor. This means that the use of any type of mechanism used to generate structures by programming can be considered.

Furthermore, since the internal format of 4D structure definition XML files is public (see the section “Format of a 4D Structure Definition” on page 331), it is possible to build this type of file from other database environments or any design application in order to generate 4D databases automatically.

► To create a database from a structure definition:

1 **Select the New > Database From Structure Definition... command in the File menu of 4D.**

A standard Open document dialog box appears so that you can specify the definition file to be opened. You must select an XML format file that respects the “grammar” of 4D structure definitions (the program validates the file via the DTD).

2 **Select a structure definition XML file then click OK.**

4D displays a dialog box that can be used to choose the name and location of the database to be created.

3 **Choose the name and location of the database to be created then click on Save.**

If the XML file is valid, 4D closes the current database (if applicable) and creates a new structure based on the structure definition file and displays the Explorer window. An empty data file is also created by default.

5 Creating Forms

Forms provide the interface through which information is entered, modified, and printed. A user interacts with the data in a database using forms and prints reports using forms.

Each table in your database generally has at least two forms. One form is for listing records on-screen and the other form displays one record at a time and is used for data entry and modification. The form that lists records is called the *output form* or *list form* and the form that displays one record at a time is called the *input form* or *detail form*. When you are viewing records using the list form, you can double-click a record to view it using the current detail form.

This chapter covers the following topics:

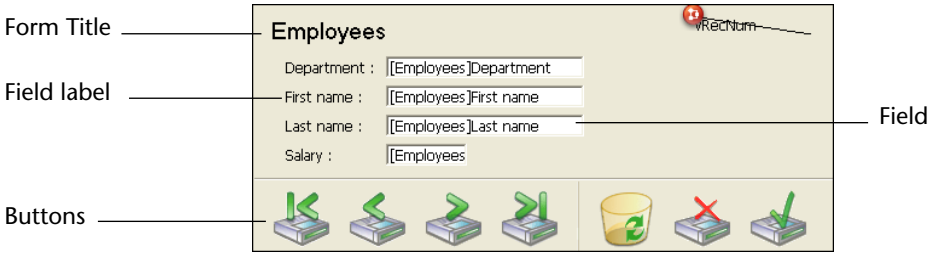
- Input and output forms
- Project forms and table forms,
- Creating forms for data entry and display,
- Establishing default input and output forms,
- Deleting forms,
- Renaming forms.

For information about customizing forms, see the chapter 6 “Form Editor” on page 371 and chapter 7 “Fields and Active Objects in Forms” on page 483. For more information about user forms, please refer to the chapter “User Forms” on page 685. For information on creating and customizing list forms for listing records on-screen and for printing see chapter 8 “Output Displays and Reports” on page 635.

About Forms

The form is the interface object that you use for data entry, for listing records, for printing reports and mailing labels[1], and (in custom applications) for custom dialog boxes and palettes.

4D lets you create standard forms quickly. It also provides powerful tools that let you create forms that implement sophisticated interfaces. Your forms can provide exactly what your database needs. With only point and click operations, you can create a basic form as shown below.



4D has two tools for creating and modifying forms, the Form Wizard and the Form editor.

The Form Wizard

The Form Wizard is your starting place for creating any type of form. With the Form Wizard, you can create a new form by choosing the desired fields from a list and the desired form template from a drop-down list. Form templates control the appearance of forms. A template specifies such characteristics as form size, platform interface, font attributes, and buttons.

1. You can also print reports and labels with the Quick Report and Labels editors. These editors can be used in the Application environment.

The Form Editor

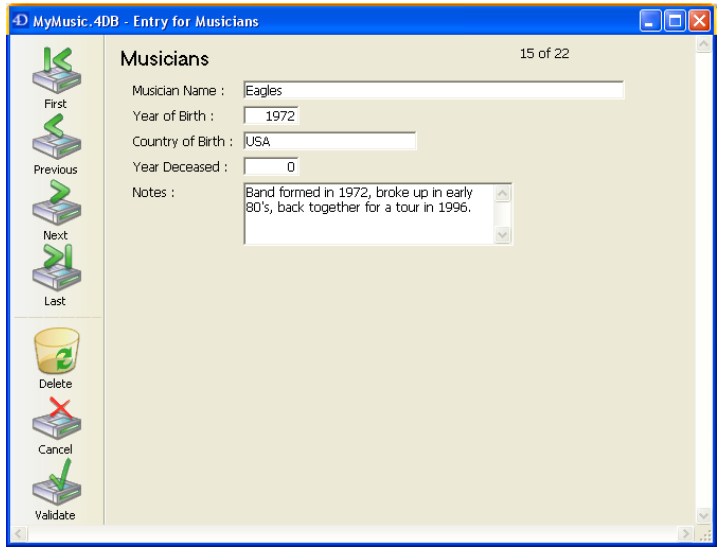
The Form editor is an object-oriented drawing environment that lets you customize forms by manipulating objects on the form directly. For example, you can reposition objects, add objects not supported by the Form Wizard, create multi-page forms with tab controls, enforce business rules by specifying data entry constraints, specify form access privileges, associate a custom menu bar with a form, and write form and object methods that run automatically when the form is used.

This chapter discusses creating forms for data entry and display using the Form Wizard. Chapters "Form Editor" on page 371 and "Fields and Active Objects in Forms" on page 483 discuss the Form editor.

Input and Output Forms

Input and output forms provide access to the information in your database. You use forms to enter, view, modify, and print information.

Input forms display one record at a time. They have enterable areas for fields and buttons for working with records:



Output forms display several records as a list. Each row is a record and each column is a field. Typically, only a few fields appear in the output form. Here is an output form that was created using a standard form template:

Album Title :	Musician :	Format :
Rhapsody in Blue, An American in Paris	George Gershwin	CD
Sound of Jazz	Lionel Hampton	CD
Nat King Cole's Greatest Love Songs	Nat King Cole	CD
The Best of the Sylistics	Stylistics, The	Cassette
Johnny Mathis, 16 Most Requested Songs	Johnny Mathis	CD
Best of B. B. King	B. B. King	DVD
Carpenters - Their Greatest Hits	Carpenters, The	CD
Jazzis Magazine April 1995 Collection	Various	CD
Virtuoso - Ludwig Van Beethoven	Berliner Philharmoniker	CD
Temptations 25th Anniversary Volume II	Temptations, The	CD
Brahms Piano Quintet - Clarinet Quintet	Benda Musicians, The	CD
Best of Gladys Knight & the Pips, 1973-1988	Gladys Knight & the Pips	Cassette
Bad	Michael Jackson	Video
Double Good Everything	Smokey Robinson	CD
Gettin' Ready	Temptations	CD
The Long Run	Eagles	CD
Kool & The Gang Spin Their Top Hits	Kool & The Gang	CD
Lucille and Other Classics by Kenny Rogers	Kenny Rogers	CD

Although the above example shows the typical use of a form, any form can be designated as either an input or output form.

For instance, a form that contains several fields set out vertically and one or more picture field(s) can be used perfectly well as an output form.

When a form is designated as an output form, it displays several records as a list that you can scroll through. You use an output form to do the following:

- Scroll a list of records,
- Highlight a subset of records,
- Double-click a record to display it in the input form,
- Enter or modify records directly in the list.

Input forms display one record at a time. You can enter and modify information, move from one field in the form to another, and move from one record to another. An input form usually has buttons that you use to:

- Save a new record to disk or save modifications to an existing record,
- Cancel changes to a record,
- Navigate from one record to another,
- Delete a record.

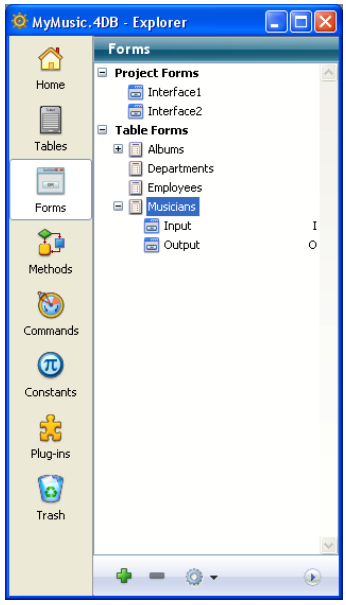
To find out more about designating default input and output forms, please refer to the paragraph "Designating the Input and Output Forms" on page 368.

Table Forms and Project Forms

4D lets you create two categories of forms: **table forms** and **project forms**. Basically, table forms are attached to specific tables and thus benefit from automatic functions useful for developing applications based on databases. Project forms are independent forms that are not attached to any table. They are intended more particularly for creating interface dialog boxes as well as components.

Note Project forms can be used to create interfaces that comply with OS standards more easily. In particular, calling (via the DIALOG command) project forms that display the selections of records in subforms is now recommended by 4D for the display of records in list form. With a little additional programming, this combination is an improvement on the MODIFY SELECTION and DISPLAY SELECTION commands.

Table forms and project forms are grouped separately on the Forms page of the Explorer:

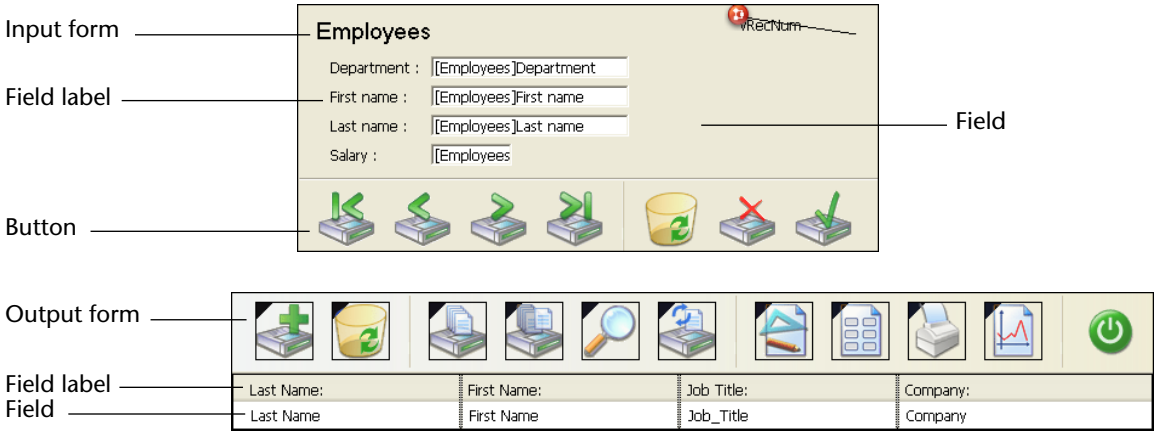


Specific Characteristics

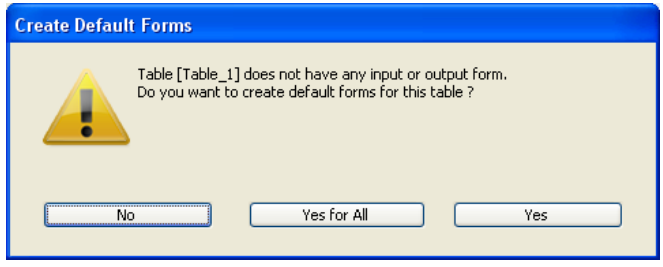
Table Forms

Every form is attached to a table. Each table must usually have at least one form so that information can be entered into fields and displayed on screen. Typically, a table has separate input and output forms. The input form is the one used for data entry. It displays one record per screen and typically has buttons for saving and canceling modifications to the record and for navigating from record to record (i.e., First Record, Last Record, Previous Record, Next Record). The output form displays a list of records, with one line per record. The results of queries are shown in the output form and the user can double-click a line in an output form to display the input form for that record.

The following illustrations show a typical input and output form:



When you try to display the records of a table before creating a form for this table, 4D asks you if you want to it to create default input and output forms for you.



Note With the Automatic Form Creation option in the Preferences, you can, for example, set 4D to automatically create default forms and therefore not display the Create Default Form dialog box. For more information, refer to the "Automatic Form Creation" paragraph on page 180.

Click **Yes** to create default forms. You can always return to the Design environment and modify them or replace them with more sophisticated forms. Without making any modifications, you can start using these forms to enter and display data in your database. You can also click **No** if you do not want to associate a form with the table. Data entry and/or display can then be carried out using project forms.

Your database can use up to 32,000 forms per table, that perform specific functions.

Project Forms

Project forms differ from table forms in the following manner:

- Project forms can only be of the detail (page) type. The mechanisms for output (list) forms are not compatible with project forms.
- Project forms do not appear in the List of tables and cannot be designated as the current input or output form. They cannot be used in the label editor nor in the 4D import/export editor.
  Project forms can only be displayed using the DIALOG command or as inherited forms (see the "Inherited Forms" paragraph on page 472).
- Project forms can contain the same types of objects as table forms, including fields.
  When fields are used, the project form stores the number of the table and the field. When a form is copied from one database to another or within a component, the references are also copied. The table and field used are those of the target database. In the case of incompatibility (non-existent table, incorrect field type, etc.), the form will not work correctly.
  Since project forms are mainly intended to be used in the context of the DIALOG command, the buttons with standard actions for record management (Next record, Delete record, etc.) are not provided by default in the editor nor in the New Form Wizard. You must manage the display of records and any data modification therein using language commands.
  On the other hand, when project forms are used as inherited forms by the table forms, the use of automatic record management mechanisms is possible.
- Project forms can have a form method, like table forms, which can be accessed from the "Methods" page of the Explorer.

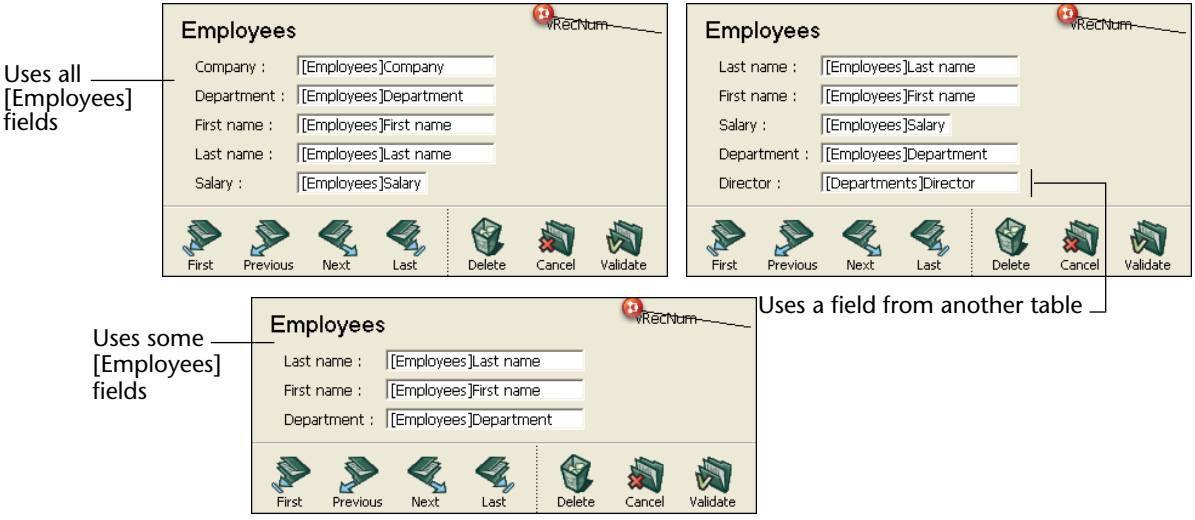
General Characteristics

In custom applications you can use the language to control which forms to use. For example, you may want to switch sets of forms depending on whether the user is using a monochrome or color monitor. You can also use the language to use different sets of forms for Web browsers and 4D Client users. When you write a custom application, you can create forms for use as custom dialog boxes or floating palettes. In custom applications, you can also use multiple processes to allow users to work with several forms simultaneously.

A form can display fields from more than one table. You can place fields from a related One table on a form and allow users to enter values directly into the related One table. You can also include a *subform* that displays a list of records from a related Many table. A subform displays a list of records from another table or a subtable in the master table. With a subform, the user can view, enter, and modify records in another table. This is sometimes known as a *master-detail relationship*. For example, an invoicing application would use a subform on the invoicing input form that lets the user enter line items for the invoice. Although the line items appear on the invoicing screen, the line item records are actually stored in a related Many table.

A form used for data entry can have more than one subform. For example, a contacts manager database can use one subform for telephone numbers, another subform for ToDo's, and another for prior contacts with the person. Each subform displays records from a different related Many table.

A particular form can use some of the fields in a table or all of the fields. You might have two input forms, for example — one for use by a clerk and another for use by supervisors — neither of which contains all the fields. You might use another group of fields for the screen display and yet a fourth group for a printed report.



Forms can be modified at any time, regardless of whether you have entered data into the database. Changes to a form do not affect the data stored on disk in any way.

Each form has one or more display pages in which fields and other enterable objects appear. If your fields don't fit on one page, you can create additional pages. When you create a multi-page form, you also add buttons or a tab control to allow users to move from one page to another.

Each form also has a background page (a page zero) on which you place objects that appear on all display pages. Use the background page to place background graphics, buttons, a tab control, and other graphic objects that define the "look" of the page, such as rectangles and labels.

Note When a multi-page table form is used as an output form (e.g., when it is printed), only the first display page appears.

Transforming a Table Form into a Project Form (and Vice Versa)

It is possible to change a table form into a project form (or to carry out the opposite transformation) at any time.
Be careful, when transforming table forms into project forms, any automatic functioning concerning data management that is present in the table form will no longer work once the form has been transformed. Similarly, a "list form" or "list form for printing" type form will be transformed into a project form of the "page" type.

Changing the type of a form can be done by drag and drop or copy/paste on the **Forms** page of the Explorer. This can be done in the same database or between two different databases.

► To transform a project form into a table form and vice versa:

1 **On the Forms page of the Explorer, click on the form that you want to transform and drop it on the destination item.**

When transforming a project form into a table form, you must drop the form onto the name of the table to which it will be attached.

By default, the form is moved when the drag and drop operation is carried out inside the same database. If you want to copy the form, hold down the **Alt** (Windows) or **Option** (Mac OS) key during the drag and drop. When the drag and drop is between two different databases, the form can only be copied.

You can also use the standard **Copy/ Paste** commands in the contextual menu of the Explorer.

Active Objects and Graphic Objects

There are two kinds of objects in a form: active objects and graphic objects.

Active Objects

Active objects perform operations on data or provide a custom user interface. Active objects include the following:

- Fields,
- Variables, in other words enterable and non-enterable areas, for entering or displaying data
- Buttons (standard, 3D, Highlight, Invisible, or Picture) that perform actions
- Tab controls
- Combo boxes
- Pop-up menus and drop-down lists
- Hierarchical pop-up menus and hierarchical lists
- Radio buttons and check boxes used for entering values into Boolean fields or variables
- Thermometers, rulers, or dials that show relative values
- Lists and hierarchical lists, that allow the user to select from the list or drag items to or from the list
- List boxes, used to display data as columns and rows,
- Graph and Plug-in areas
- Splitters for resizing form areas
- Subforms that display data from other tables and subtables.

Graphic Objects

Graphic objects are geometric or textual elements that enhance the appearance of the form. Graphic objects include the following:

- Rectangles, ovals, and circles for enhancing the appearance of a form
- Text, for labelling areas in the form
- Grids, for aligning other objects

- Graphics from applications other than 4D, including pictures stored in the Picture Library.

All objects, whether active or graphic, are handled in the same way in the Form editor — they are created by being drawn or dragged and dropped; they can be selected and moved or resized; they can be duplicated, cut, copied, and pasted; they can be aligned to each other or to an invisible grid, and their appearance can be changed.

Object Properties

Each object has a set of properties. For graphic objects, properties include foreground and background color, line width and fill pattern, resizing and repositioning options, and font size and attributes. For active objects, properties may also include the object's relationship to data, the object's "action" when it is used, a help tip, and the object's method. When the Form Wizard generates a form, it assigns appropriate default properties to both graphic and active objects[1]; you can modify these properties in the Form editor.

Graphic objects have no impact on the data. You can create a graphic object on a form simply by drawing it and making any necessary modifications to its appearance. For example, when you create a text area, you draw the area, then you type the text you want to display. You can change the text at any time without affecting the data.

Active objects require instructions about their relation to the data or instructions on the actions that they are to perform. In simple cases, the Form Wizard does everything for you. Entry areas for fields are automatically associated with the appropriate field in the database structure and buttons automatically perform the appropriate actions. In other cases, you can specify additional instructions in the Form editor using the Property List of the object.
For example, you can assign special instructions to a button by writing a method. The method remains attached to the object as one of its properties. If the object is copied and pasted, it retains all its properties, including its method.

1. For example, the Form Wizard generates buttons that perform standard actions and assigns appropriate resizing and repositioning options to decorative rectangles.

Chapter 6 “Form Editor” on page 371 provides detailed information about working with the Form editor. Chapter 7 “Fields and Active Objects in Forms” on page 483 provides complete information about active objects.

Object Libraries

Object libraries can store any type of active or graphic objects: buttons, text, pictures, hierarchical lists, etc. The objects are stored with all their properties, including, for active objects, their associated methods and standard actions.

Objects stored in a library can be reused in any form by using copy-paste or drag-and-drop operations. They are a kind of permanent clipboard.

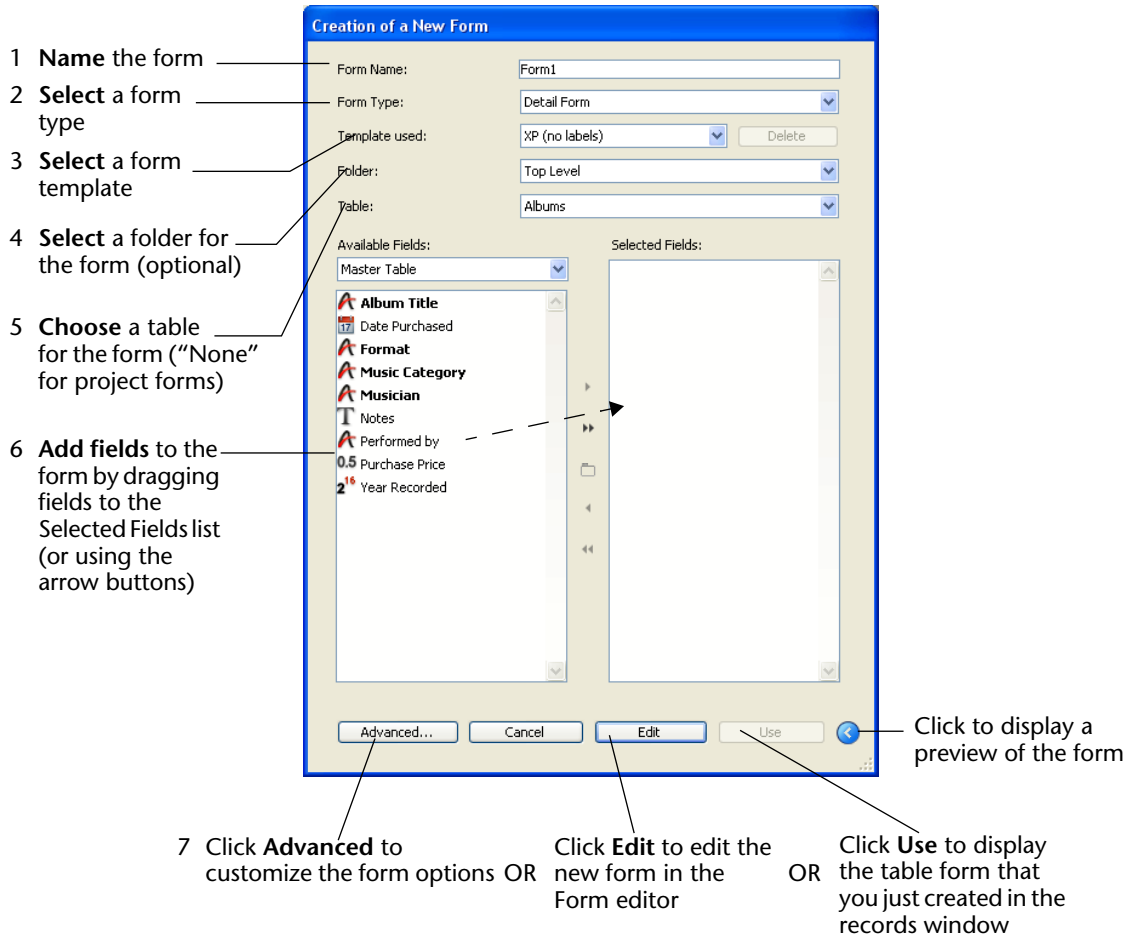
Using libraries, you can make up backgrounds of form objects grouped by graphic families, functionalities, etc. These libraries are stored as external files which makes it easy to use them in different databases.

Creating and using object libraries is detailed in the “Using Object Libraries” paragraph on page 461.

The Form Wizard

You can create new forms quickly with the Form Wizard. You can use a new form immediately after creating it or choose to edit the form using the Form editor.

The Form Wizard has two screens. The Basic screen lets you create new forms with a few simple operations. The steps for creating a form are shown in the following illustration.



For complete information on using the Form Wizard, see the section "Creating a New Form using the Wizard" on page 350.

The Form Editor

The Form editor is a powerful object-oriented drawing environment. Each of your forms can be displayed in a separate window and several forms can be open the same time. Objects on the form can be created or manipulated with the tools of the Form editor. You can set each form's properties and each form object's properties as well. While you are designing a form in the Form editor, you can test the form without having to first close the Form editor.

The Form editor offers many customization options that are not available in the Form Wizard, including the ability to:

- Set access privileges for the form,
- Assign properties to each object,
- Attach a method to the form and to each form object,
- Add pictures from the Picture Library to the form,
- Resize and reposition each object directly,
- Add types of interface objects that are not supported by the Form Wizard, including tab controls, drop-down lists and pop-up menus, picture buttons, list boxes, combo boxes, check boxes and radio buttons, hierarchical menus and hierarchical lists, scrollable areas, plug-ins, graphs, splitters, and additional subforms.[1]
- Set drag-and-drop properties for individual objects,
- Set data entry controls such as minimum, maximum, and default values, entry filters, and choice lists,
- Customize automatic resizing and repositioning options for each object,
- Add objects from an object library to the form.

For a complete description of the Form editor, see the chapters "Form Editor" on page 371 and "Fields and Active Objects in Forms" on page 483.

1. The Advanced options in the Form Wizard only lets you add one subform.

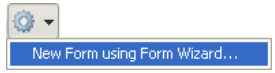
Creating a New Form using the Wizard

This section gives the basic steps for creating a form using the Basic screen of the Form Wizard. For information on the Advanced options of the Form Wizard, see "Using the Form Wizard's Advanced Options" on page 355.

You can also create new blank forms that you can build directly in the Form editor. For more information about this, refer to the "Creating a Blank Form Using the Explorer" paragraph on page 367.

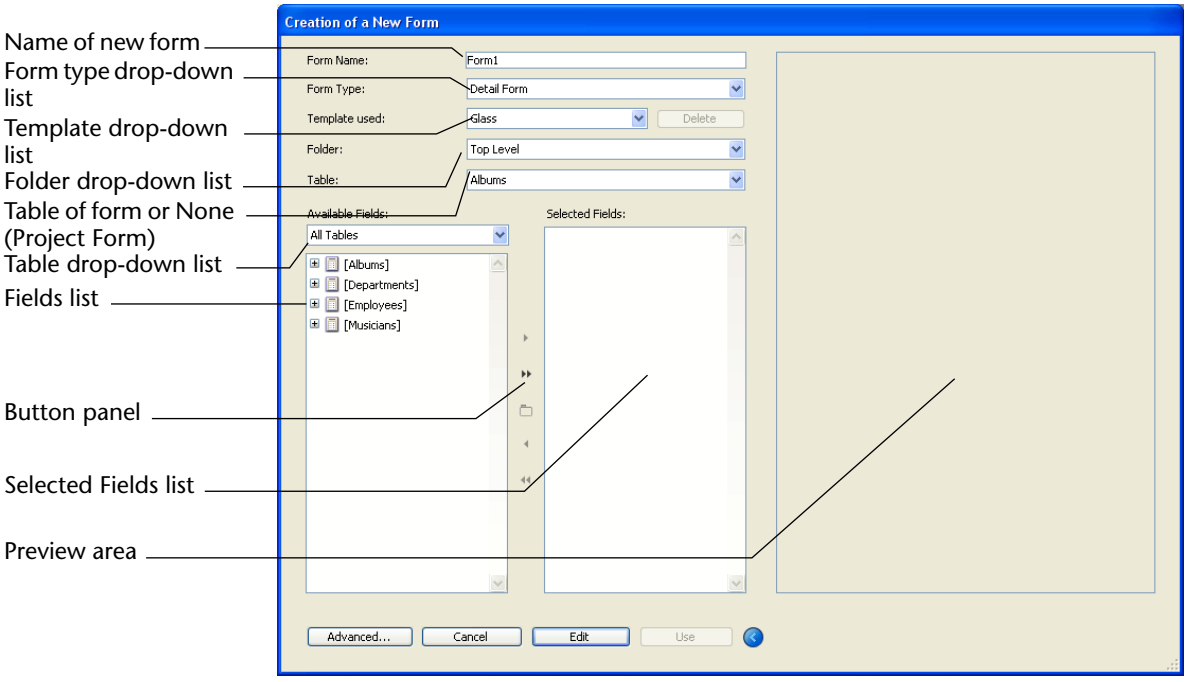
► To create a new form using the Form Wizard:

1 **Choose New > Form... in the File menu or using the New button of the toolbar.**
OR
On the Forms page of the Explorer, click on the options menu and choose New Form using Form Wizard.



Note The contents of the options menu vary depending on the item selected in the Explorer.

4D displays the Basic screen of the Form Wizard.



2 **To create a table form, choose the table for the form from the "Table" menu.**

Its fields are then listed in alphabetical order in the Available Fields area when the Master Table option is selected in the drop-down list.

OR

To create a project form, choose None (Project Form) from the "Table" menu.

The list of fields from all the tables will then be displayed in the Available Fields area. The **List Form** and **List Form for Printing** types are removed from the list of available form types (see below).

Note For more information, please refer to the "Table Forms and Project Forms" paragraph on page 339.

3 **Name the form by filling in a name in the Form Name area.**

You can refer to the form by name using the language.

4 **Choose a Form Type from the Form Type drop-down list.**

Your choices are:

- **Detail Form**: A form for data entry and modification.

- **List Form** (table forms only): A form for listing records on the screen.
- **Detail Form for Printing**[1] : A printed report with one page per record, such as an invoice.
- **List Form for Printing**[1] (table forms only): A printed report that list records.

5 Choose a template for the form.

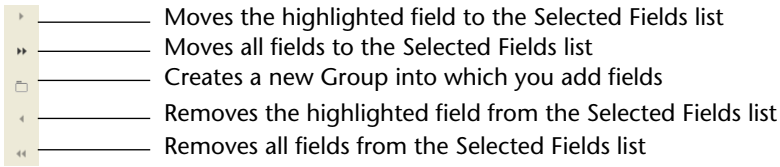
The template controls several aspects of the form appearance, including font attributes, field label placement, the design of decorative rectangles surrounding fields, and platform interface. 4D ships with several templates and you can use the Form Wizard to add custom templates to this list. For more information about adding custom templates, see the section "Creating a Form Template" on page 366.

6 (Optional) Select a storage folder for the form.

If you select a folder name from the drop-down list, the form will be placed in this folder. Folders can be used to organize the objects of your applications and are managed on the Home page of the Explorer. For more information, refer to the "Home Page" paragraph on page 103. By default, the form is created at the Top Level, i.e., not in any storage folder.

7 Select the fields you want on your form from the "Available Fields" area.

You can double-click on the fields, use drag and drop or use the buttons in the central panel:



You can select any fields of any type, except BLOB fields. You can create forms that include fields from:

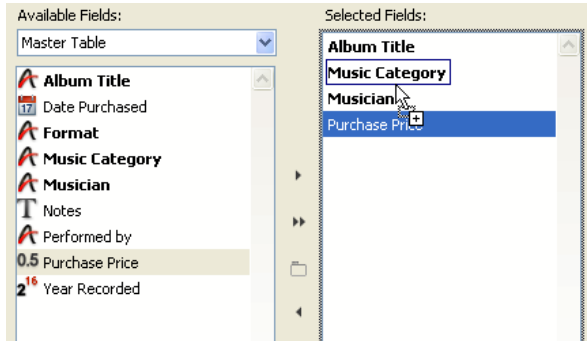
- The master table (in the case of table forms),
- A related One table,
- Any table.

1. For information on creating forms for printing, see Chapter 7.

The Subform page in the Advanced options screen lets you create subforms that display fields from Many tables as well as tables that are not related.

If you select fields from a table that is not the master table or an automatic related One table, you will need to use the language to manage data entry and display in the fields you select.

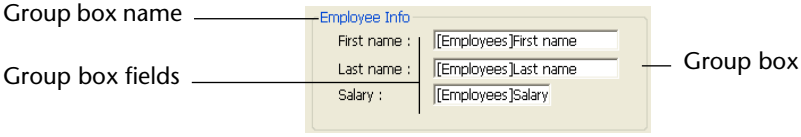
When you add fields or change the type or template of the form, your changes are shown in the preview area on the right-hand side of the wizard. After placing fields in the form, you can change their order using drag and drop in the Selected Fields area:



Note If your screen is too small for you to create a form big enough to contain all the selected fields, the wizard creates a multi-page form and place all the buttons and static objects on page 0. When you edit this type of form, you will first need to display the page 0 in order to be able to edit these objects.

8 (Optional) If necessary, create one or more group boxes in the Selected Fields list.

A group box has its own label and a set of fields. It looks like this:

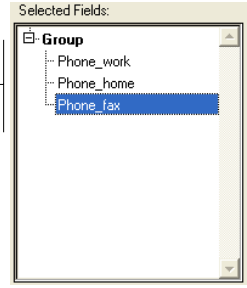


To make a group box, click on the group box button . An item is added in the Selected Fields list. Its default name is “Group.” You can rename it as necessary.

Note The name of a group box is static text; you can use a “localizable” reference as with any 4D label (see Appendix C on page 1293).

You can add fields to the group box by simply dragging them over from the Available Fields list and dropping them onto the new item. The Selected Fields list takes the form of a hierarchical list. As each field is added to the group box, it appears below the group box name.

Fields in the group



Note When a group box item is expanded, you can add fields to it that are already in the Selected Fields list by drag and drop.

9 If you want to edit the new form in the Form editor, click Edit.
OR
If you want to test the form, click Use (table forms only).
OR
If you want to customize the new form with the Form Wizard’s advanced options, click Advanced.

Using the Form Wizard's Advanced Options

To display the advanced options of the Form Wizard, click on the **Advanced...** button on the basic screen of the Wizard. If you want to go back to the first screen of the Form Wizard, you can do so by clicking the < **Back** button.

The Advanced screen also lets you create new forms with point-and-click operations but offers a wider variety of customization options.

The customization options depend on the form type that you select in the first screen of the Form Wizard. The Form Wizard supports the following form types:

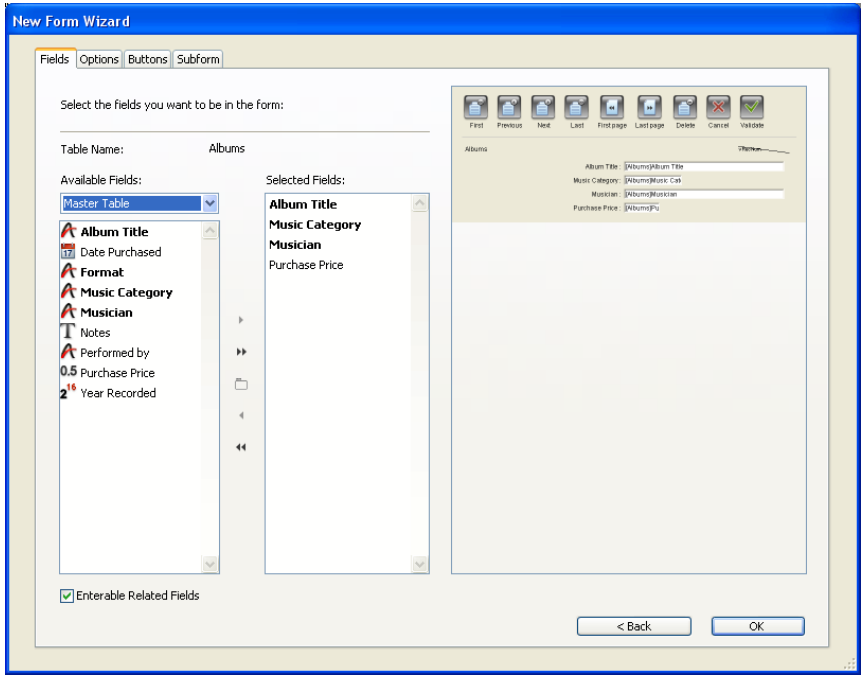
- Detail Forms
- List Forms (table forms only)
- Detail Forms for Printing
- List Forms for Printing (table forms only).

In addition, the Form Wizard lets you save your customization options as a template. The template name is added to the Template drop-down list that appears on the Basic screen of the Form Wizard. With user-defined templates, you can create highly customized forms quickly from the Basic screen of the Form Wizard simply by selecting the desired fields and your custom template.

This section describes the advanced options available for creating Detail forms. For more information about the creation of List forms, refer to chapter "Output Displays and Reports" on page 635.

Fields Page

The Fields page is similar to the Basic screen of the Form Wizard.



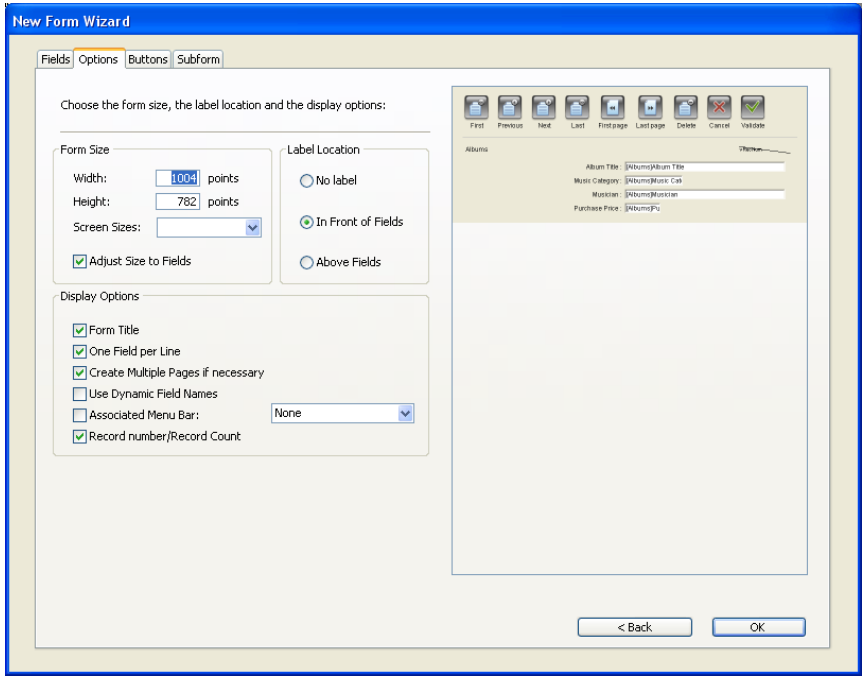
The Fields page is used to add fields to the form in exactly the same way as on the Basic screen of the Form Wizard. This functionality is duplicated for users who want to skip the Basic screen and go directly to the Advanced options screen. For a complete description of the process of adding fields to the form, see the section “Creating a New Form using the Wizard” on page 350.

Enterable Related Fields

The Fields page offers an additional option: **Enterable Related Fields**. This option lets you choose whether or not to assign the “Enterable” property to the fields of related tables. By default, this option is checked (the fields are enterable). It may be useful to uncheck this option, for instance if you want to prevent users from being able to modify the values of related fields when the “Auto assign related value in subform” option is checked for a relation (for more about this, refer to the “Setting Relation Properties” paragraph on page 308).
For more information about the “Enterable” property, refer to the “Setting the Enterable and Mandatory Attributes” paragraph on page 490.

Options Page

The Options page is used to set various options concerning form size, placement of field labels and the display of the form.



Form Size

This area lets you specify the form size. You can adjust the form size to its contents or set the form to a fixed size either by entering its maximum width and height, or by selecting a screen size. You can also combine the two settings.

The Screen Sizes drop-down list gives you the following choices:

- Automatic (the form size adapts to the current screen)
- Low Resolution 640x480
- Medium 800x600
- Standard 1024x768
- High resolution 1280x1024
- Large screen 1600x1200.

Note The values correspond to the *width* x *height* ratio, expressed in pixels.

When you enter a screen size or choose a size from the drop-down list, the preview area changes to reflect your selection. The Form Wizard will try to adjust field and object placement on the form so that all the form objects will fit in the selected screen size. If the **Create Multiple Pages if necessary** option is selected and 4D cannot make all the fields fit on one page, it will generate multiple display pages to fit all the fields on the form. For more information about this option, refer to "Display Options" on page 358. If the Form Wizard generates multiple pages, it places buttons, the form title, and decorative rectangles on the background page (page 0).

- **Adjust Size to Fields**: If you check this box, the Form Wizard will shrink the background items around the fields so that less blank space is left.

Label Location

The Label Location area on the Options screen allows you to control where a field label is placed in relation to the field. If you want labels, they can be placed either in front of or on top of the fields.

Display Options

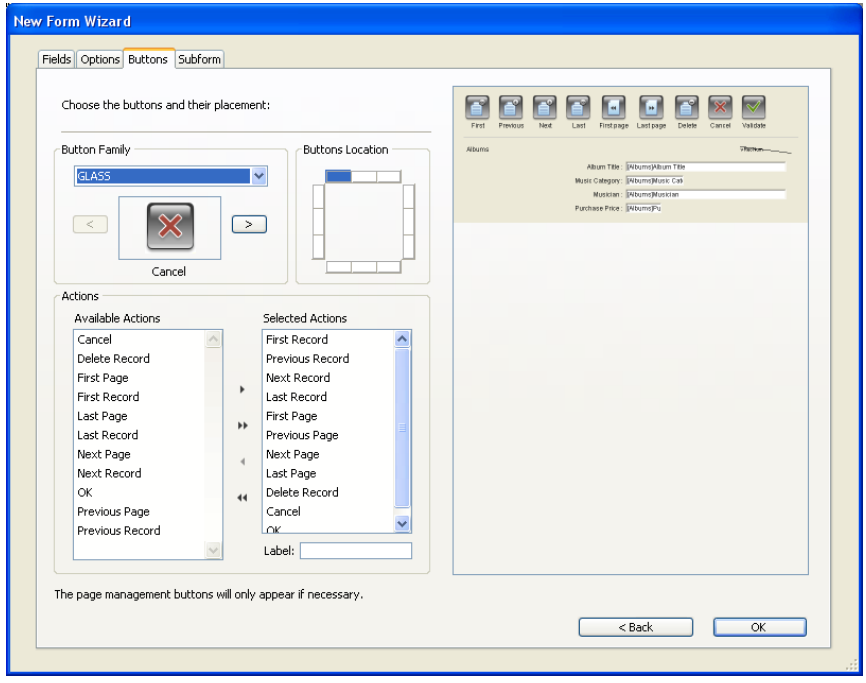
The Display Options area on the Options screen lets you add several optional elements to the form and set additional options. Your choices are:

- **Form Title**: Adds the name of the table as the title of the form above the fields. This option is not available for project forms.
- **One Field per Line**: Check to arrange the fields vertically. If this option is not checked, the Form Wizard will try to arrange fields in rows.
- **Create Multiple Pages if necessary**: Check to have the Form Wizard create extra pages automatically if the fields do not fit on one page. If you use this option, the Form Wizard places the appropriate objects on the background page.
- **Use Dynamic Field Names**: When this option is selected, field and table names are inserted in the form as dynamic references. This ensures that the field and table labels will reflect any changes made to the field or table name. Table or field names can be modified in the Structure editor or using the SET TABLE TITLES or SET FIELD TITLES commands. For more information, see "Inserting Dynamic Table and Field Names" on page 487.

- **Associated menu bar**: Check this option and select the name of the menu bar that you want to associate with the form. For more information, please refer to "Assigning a Menu Bar to a Form" on page 404.
- **Record Number/Record Count**: Adds a 4D variable (named *vRecNum* by default) to the form that display the current record number and the total number of records. This option is not available for project forms.

Buttons Page

The Buttons page lets you customize the buttons used in the form.



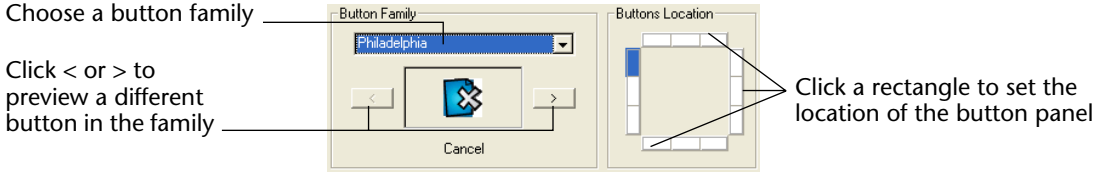
Detail forms use buttons that allow the user to save and cancel changes to a record, or move from one page to another in a multi-page form. In addition, table forms can have buttons that can be used to move from one record to another (first record, last record, next record, previous record), to add or delete records in a subform, or to delete the current record.

On the Buttons page, you can choose a button design, choose the desired button actions, specify the position of the buttons on the form, and label each button.

Note In the Form editor, you can add, delete, or reposition individual buttons and attach a method to a button that specifies its action when clicked. For more information, see "Buttons" on page 551.

Family and Location

The Button Family and Buttons Location areas let you choose the style and location of the buttons.



Choose a family from the Button Family drop-down list and click the > or < buttons to preview each button.

Automatic Button Actions

4D provides a set of built-in button actions. When you assign a built-in button action to a button, you don't need to write a method to specify what happens when a user clicks the button.

The number of built-in actions will depend on the form category (table or project). For example, actions for moving between records in a table cannot be used with project forms.

For detail forms, the following built-in button actions are available via the Form Wizard:

- **OK**: Saves a new record or saves changes to an existing record,
- **Cancel**: Discards the new record or discards changes to an existing record,
- **Next Page**, **Previous Page**, **First Page**, **Last Page**: Displays the requested page in a multi-page form.

The following actions are only available for table forms:

- **Next Record**, **Previous Record**, **First Record**, **Last Record**: Saves the current record and displays the requested record,

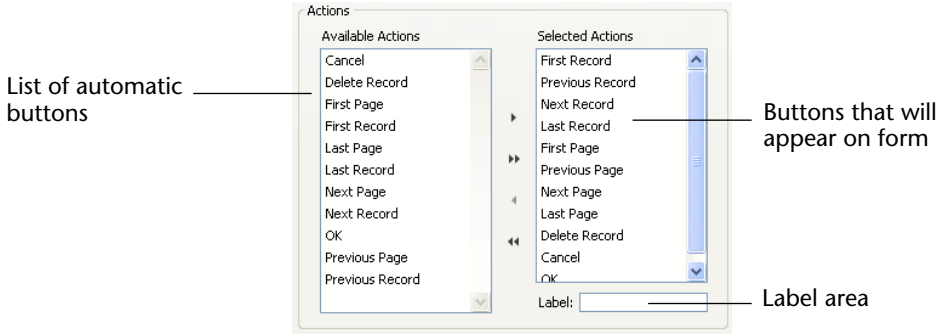
- **Delete Record**: Deletes the current record from the database (a confirmation dialog box appears).

Note When you insert a subform, 4D can automatically insert two additional subform buttons if you click the **Addition and Deletion Buttons** check box in the Options area of the Subform page. The subform buttons are: **Add** (adds a new record to a Many table or a subtable), and **Delete** (deletes the currently selected record in the subform).

4D provides other predefined actions to buttons. These actions are available when you create a form using the Form Wizard or when you modify a form using the Form editor.

For more information, refer to the "Automatic Button Actions" paragraph on page 360.

The default buttons are listed in the Actions area of the page.

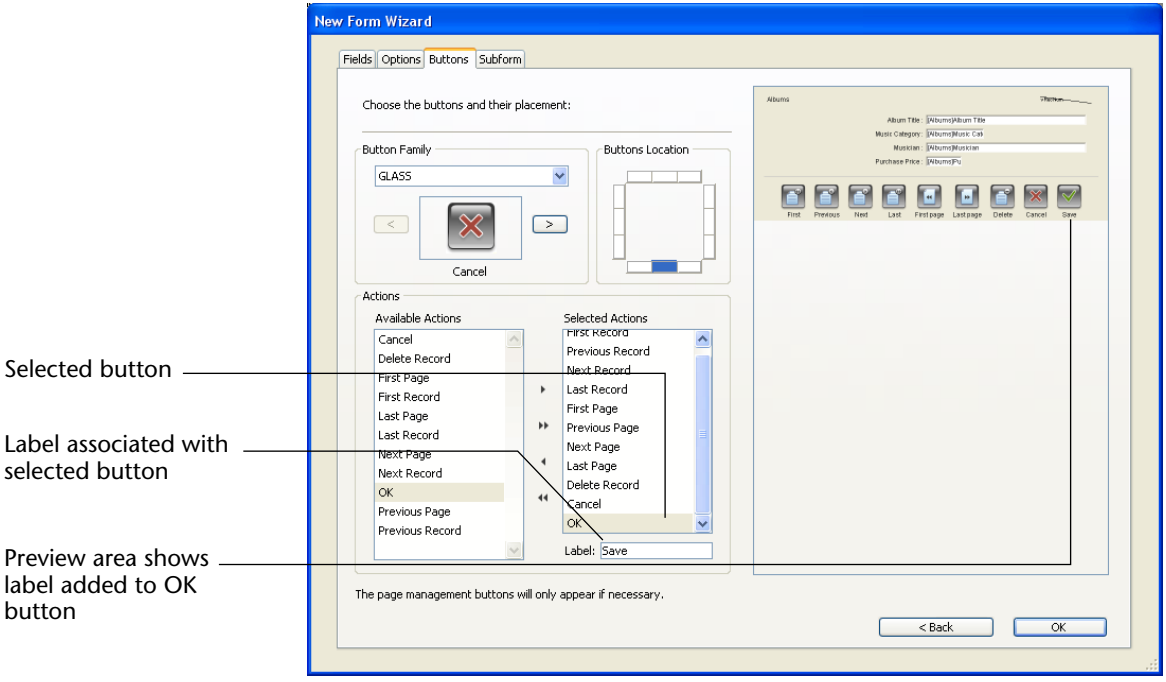


You can select and deselect automatic buttons in the same way that you can add or delete fields from the form on the Fields page. The buttons that you add to the Selected Actions area appear on the form.

Note Even when they are selected in the "Selected Actions" area, the page management buttons (Previous Page, Next Page, etc.) will only be included in the form if it is necessary to create a multi-page form.

If you want to modify the default label of a button (the template chosen must include labels), highlight the button in the Selected Actions list and enter a new label in the Label area.

The following illustration shows an **OK** button whose label has been modified.



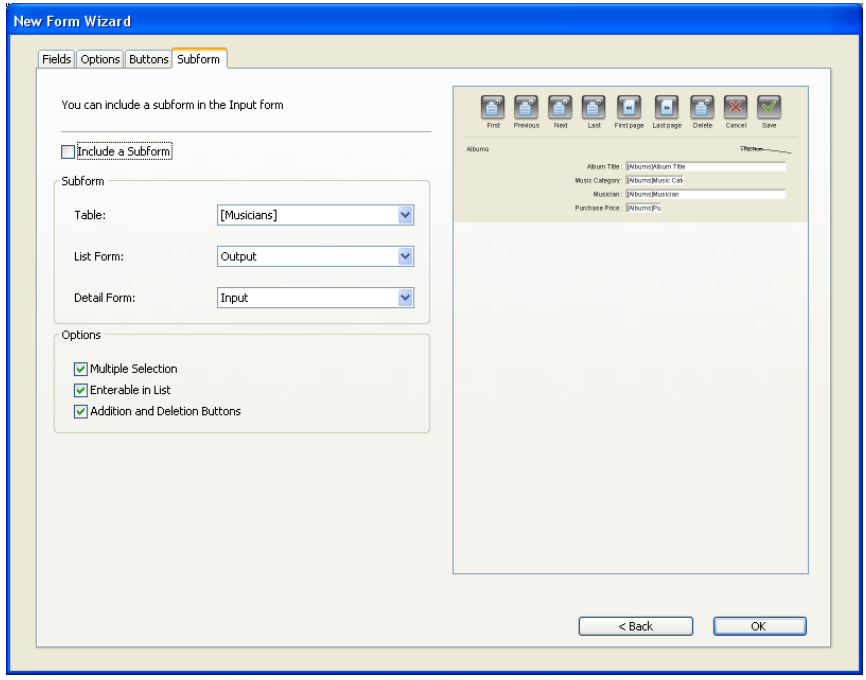
After entering the label, press **Tab** or click another button in the Selected Actions list. The label you entered is then displayed in the preview area, as shown above.

Note

- Button labels can be entered as references for the purpose of translating the database (see Appendix C on page 1293).
- Button tips are independent of button labels. If you want to assign a tip to a button, you can do so using the Property List in the Form editor. For more information, refer to "Adding Help Tips to a Field or Object" on page 517.

Subform Page

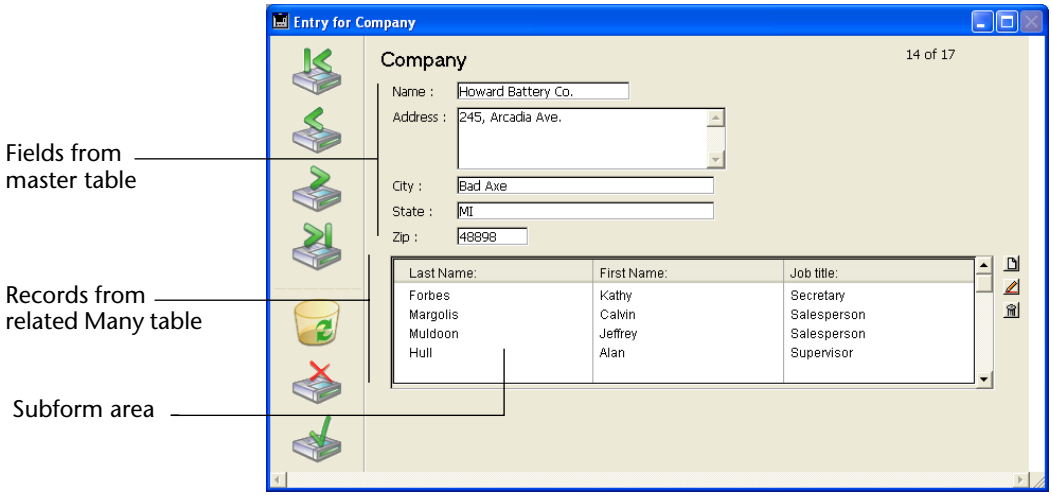
The Subform page lets you add a subform to the form. This subform must come from a related Many table if you want to be able to benefit from the automatic update mechanisms concerning subforms.



When you want to use fields from a related Many table, add a subform to the form. The subform lists several records at once.

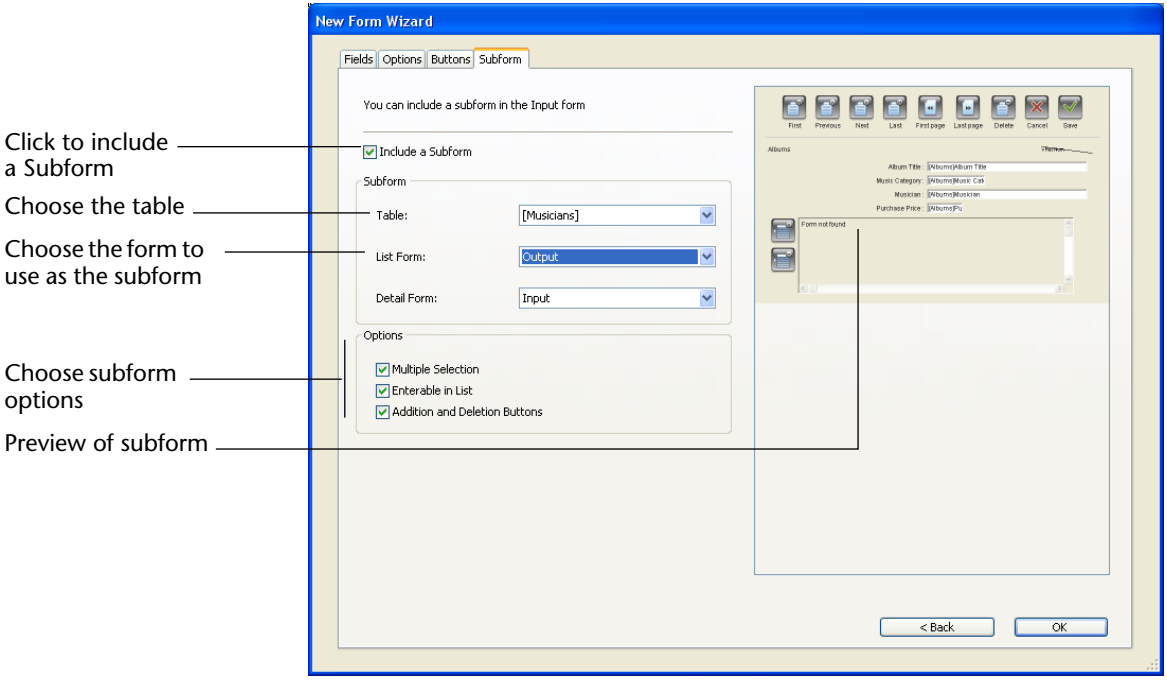
Using a subform allows you to view the related records or those from another table. You can also enter information into records that are displayed in the subform.

The figure below shows a Detail form with a subform during data entry.



You can display fields from a related Many table, or an unrelated table in a subform. If you include fields from a related Many table, the relation determines which records are displayed. If you include fields from an unrelated table or from a table with a manual relation, by default the current selection of records from that table is displayed. You can also control the selection of records using a method.

The Subform page of the Advanced screen lets you use a form as a subform, specify subform options, and add buttons to allow users to work with the subform.



To add a subform to the form, check the **Include a Subform** option.

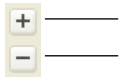
Select the table for the subform from the "Table" drop-down list and then select the subform to be used from the "List Form" drop-down list. You can also use the "Detail Form" drop-down list to set the form page to be displayed when the user clicks on the subform.

This subform then appears in the preview area as part of the current form.

You can set the following options for the subform:

- **Multiple Selection**: The user can select several subrecords simultaneously using the **Shift** and **Ctrl** (Windows) or **Command** (Mac OS) keys.
- **Enterable in List**: The user can modify subrecord values directly in the subform.

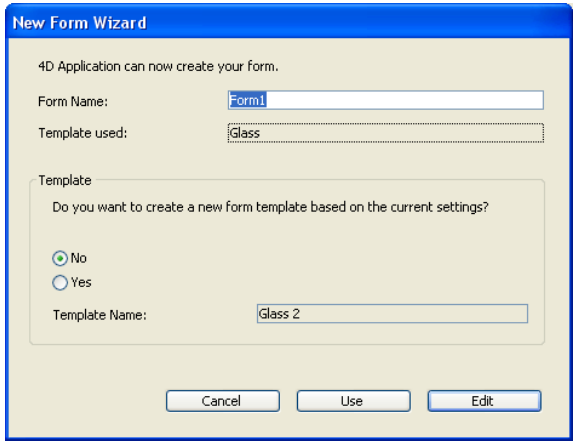
- **Addition and Deletion Buttons**: The Wizard automatically inserts two buttons in the form (corresponding to the button family specified for the form), which are associated with the **Add Subrecord** and **Delete Subrecord** standard actions:



For more information about these options, refer to the "Subforms Options" paragraph on page 629

Generating the Advanced Form

When you have finished specifying all the properties of the new form, click **OK** on any page to create the new form. When you click **OK**, the following dialog box appears:



To create the new form, click either **Use** to switch to test the form (table forms only), or **Edit** to open the new form in the Form editor.

Creating a Form Template

The form validation dialog box gives you the option of creating a new form template using the current Advanced settings by default (buttons, options, etc.). If you create a form template, its name will be added to the template drop-down list in the Form Wizard. The form template is saved separately from the form itself.

To create a form template, click on the **Yes** button and enter its name in the Template Name area.

Creating a Blank Form Using the Explorer

You can create a blank form directly without using the Form Wizard. In this case, the form is created without any fields, buttons or variables and is opened in the Form editor. It can then be put together entirely using the Form editor. Creating blank forms is useful when you want to generate dialog boxes containing only variables or plug-in areas.

Keep in mind that each form must be attached to a table, even when it does not contain any fields.

▶ To create a blank form:

1 **Display the Forms page of the Explorer.**

2 **Select the item corresponding to the category of form you want to create:**

- For a project form: select either the “Project Forms” label or an existing project form,
- For a table form: select the name of a table.

3 **Click on the add button**  **of the Explorer.**

The following dialog box appears:

Project form

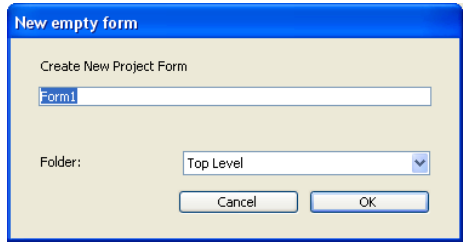
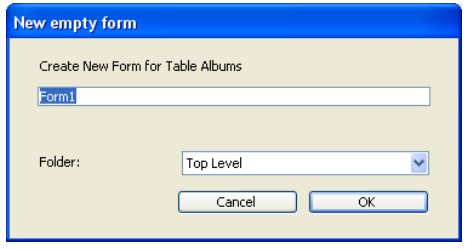


Table form



4 **Enter the form name in the corresponding entry area.**

You will use this name, more particularly, to refer to the form during programming.

5 **(Optional) Select a storage folder for the form.**

If you select a folder name from the drop-down list, the form will be placed in this folder. Folders can be used to organize the objects of your applications and are managed on the Home page of the Explorer. For more information, refer to the "Home Page" paragraph on page 103. By default, the form is created at the Top Level, i.e., not in any storage folder.

6 **Click on OK.**

The blank form is then opened in a new window of the Form editor.

Designating the Input and Output Forms

Each table has one current input form and one current output form. The input form is used for entering and modifying records, and the output form is used to list records. Usually, you use a Detail form for input and a List form for output.

You can change which form to use for input and output at any time. This change can be made in the List of tables window (see the "Browsing Different Tables and Forms" paragraph on page 947) as well as using the INPUT FORM and OUTPUT FORM commands. In this case, the changes only apply to the current worksession.

You can also specify default input and output forms in the Design environment. In this case, the changes will be saved with the database.

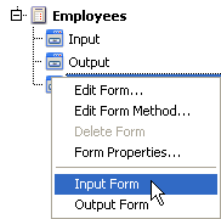
► To change the input and output forms for a table:

1 **Display the Forms page of the Explorer.**

2 **Expand the table for which you want to modify the default input or output form.**

The letter I is displayed next to the name of the current Input form and the letter O is displayed next to the current Output form.

3 **Right click on the name of the form to be designated and choose the Input Form or Output Form command in the contextual menu:**



OR
Choose the Input Form or Output Form command in the options menu of the Explorer.

Note Only table forms can be designated as input or output forms.

You can also designate the same form as the Input and Output form. In this case the character **B** (for **B**oth) will be displayed next to it.

Deleting a Form

You can delete any project form or any table form that is not designated as a current input or output form (or both). The deletion button is disabled when you select the current input or output form.

▶ To delete a form:

1 **Display the Forms page of the Explorer.**

2 **Expand the "Project Forms" theme or the table that contains the form you want to delete.**

3 **Select the form you want to delete and click the deletion button of the Explorer.**

Note You can also use the **Delete Form** command in the contextual menu of the Explorer (**right-click** on the form name).

4D asks you to confirm the deletion.

4 **Click the OK button.**

4D deletes the form. The form is moved to the Trash and can be recovered at any time so long as the Trash has not been emptied.

Note It is also possible to delete a form from the Home page of the Explorer (using the same procedure).

6 Form Editor

When you create a new form with the Form Wizard, you can choose many customization options. Using templates, you can control the appearance of fields and field labels, choose the size of the form and add a set of automatic buttons.

This is only the beginning though, since 4D provides a full-featured Form editor that allows you to modify your form until you achieve the effect that you want. With the Form editor, you can create and delete objects, manipulate objects directly, and set form and object properties.

This chapter provides an introduction to the Form editor. It covers:

- The Form editor window
- The Form editor toolbar and object bar
- The Form editor menus and tools
- The Property List window for forms and objects.

The following operations are explained in detail:

- Opening a form in the Form editor,
- Setting form properties,
- Creating and managing objects,
- Moving and resizing objects,
- Grouping and ungrouping objects,
- Aligning objects,
- Copying objects,
- Layering objects,

- Creating text areas,
- Setting object templates,
- Adding static pictures from the Picture Library or from a disk file to the form,
- Creating and using object libraries,
- Creating a multi-page form,
- Changing the data entry order of the fields,
- Working with object views,
- Setting inherited properties of a form,
- Printing a form.

The chapter "Fields and Active Objects in Forms" on page 483 covers fields and other active objects in detail.

4D Server Object locking occurs when two or more users attempt to modify the same form simultaneously. If a user is modifying a form in the Design environment, the form is locked. Other users cannot modify that same form until the first user frees the form by closing it.

Using the Form Editor

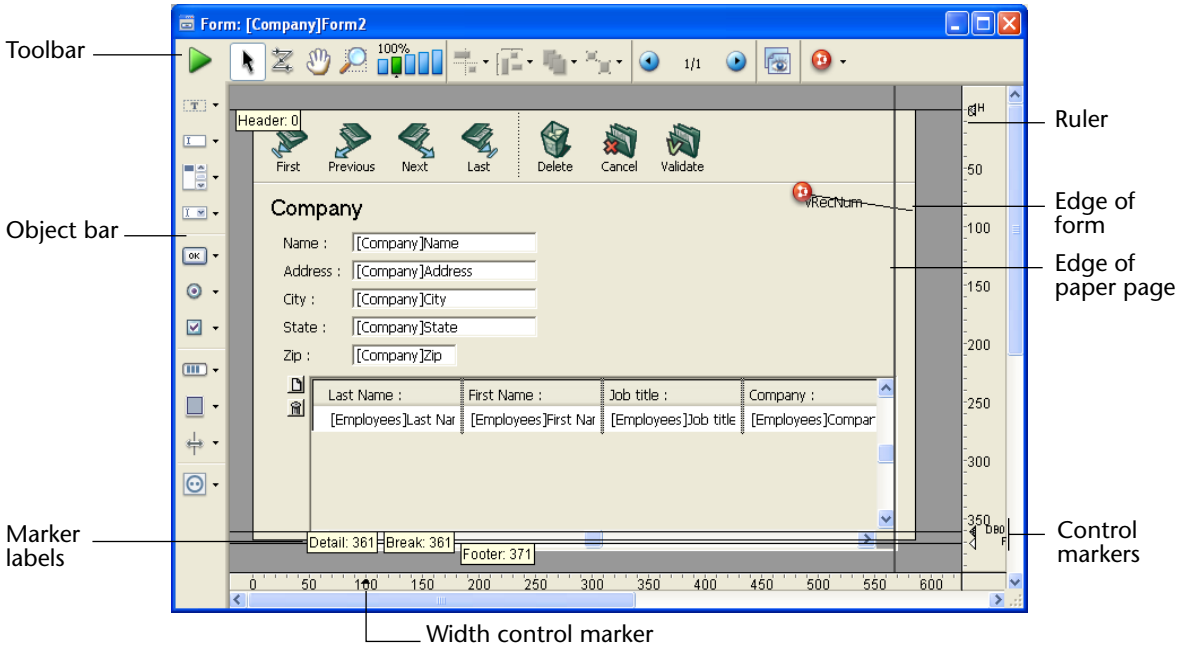
This section provides an overview of the Form editor and describes the object and tool bars. This section introduces the main components of the Form editor:

- The editor window,
- The object and tool bars,
- The editor menus,
- The Property List.

Form Editor Window

4D's Form editor is a powerful object-oriented graphics editor including special features that allow complete control over form objects.

The following illustration shows the Form editor window.



The Form editor displays each form in its own window, which has both an object and tool bar. You can have several forms open at the same time. The rulers on the side and bottom help you position objects in the form. You can change the units the ruler uses so that it measures in inches, centimeters, or pixels.

Note The output control lines are relevant only for output forms. They control the information that is listed and set header and footer areas. The label width triangle on the horizontal ruler controls the width of a label when you create a form for printing mailing labels using the PRINT LABEL command. For information on working with output control lines, see the paragraph "Moving Output Control Lines" on page 645.

The Form editor menus added to the 4D menu bar provide commands that allow you to change the data entry order of the fields, control interface elements and form pages, manage objects, and change the font and style for displaying information.

In addition to these menus, the Form editor also includes contextual menus that allow you to access numerous actions quickly. For more information, see the section "Form Editor Menus" on page 382.

Showing/Hiding Elements in the Form Editor

You can show or hide most interface elements in the Form editor. This feature allows you to show only the elements that you need to create or view in a form, or only the tools that you want to use. This option is always applied to the Form editor's current window.

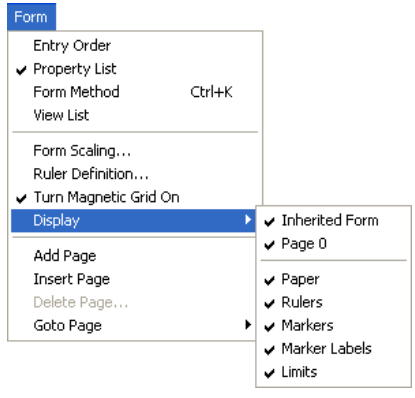
For example, it is useful to show the output control lines when you are working on an output form.

► To show or hide an element in the Form editor:

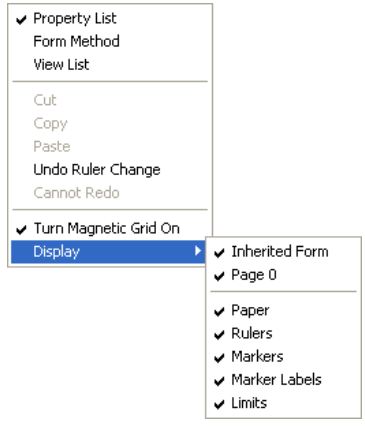
1 **Choose Display from the Form menu.**
OR
Use the Display command in the contextual menu that appears in the Form editor's window (click with the right mouse button without clicking on an object).

A hierarchical submenu appears listing all the elements that you can show or hide:

Form menu



Contextual menu



A check mark placed next to the element indicates that it will be shown. To hide an element, select the element so that the check mark disappears.

2 **Select the element that you want to show or hide.**

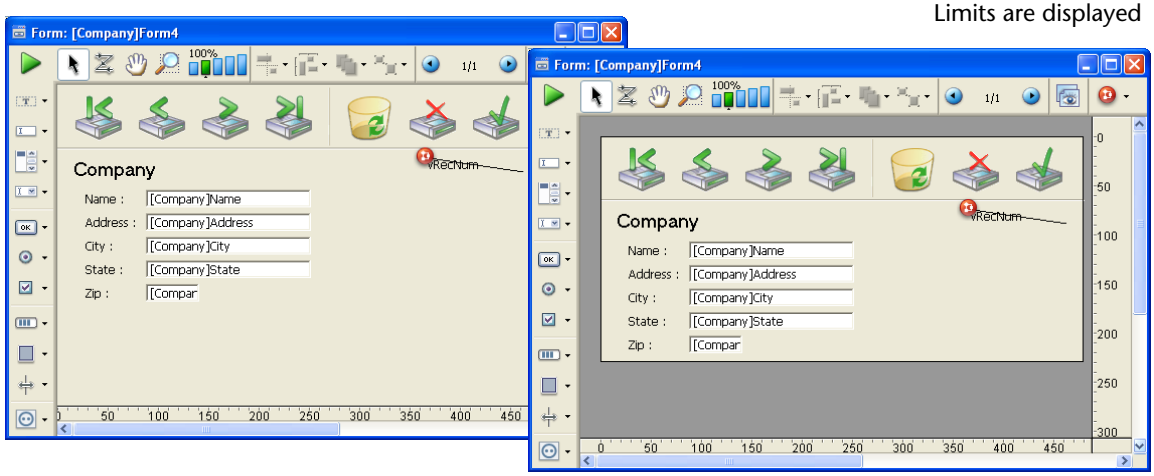
Here is a description of the commands in this menu:

- **Inherited form**: Shows or hides inherited form objects (if there is an inherited form) on the current page of the form. For more information, please refer to "Inherited Forms" on page 472.
- **Page 0**: Shows or hides the objects from page 0 on the form's current page. This option allows you to distinguish between the objects on the form's current page and those on page 0. For more information about pages in forms, refer to the section "Creating a Multi-page Form" on page 467.
- **Paper**: Shows or hides the borders of the printing page, which are shown as gray lines. This option may have no apparent effect when the **Limits** (see below) option is selected. If the size of the form is smaller than the printing page, the page's borders are shown outside of the form's viewing area and therefore do not appear.
- **Rulers**: Shows or hides the rulers in the Form editor's window.
- **Markers**: Shows or hides the output control lines and associated markers that show the limits of the form's different areas.
- **Marker Labels**: Shows or hides the marker labels, available only when the output control lines are displayed. For more information, please refer to the "Moving Output Control Lines" paragraph on page 645.

- **Limits**: Shows or hides the form's limits. When this option is selected, the form is displayed in the Form editor as it appears in Application mode. This way you can adjust your form without having to switch to the Application mode in order to see the result.

Limits are not displayed

Limits are displayed



Note The **Size Based on**, **Hor margin** and **Vert margin** settings of the form properties affect the form's limits. When using these settings, the limits are based on the objects in the form. When you modify the size of an object that is located next to the form's border, it is modified to reflect that change. For more information on form properties, refer to "Setting Form Properties" paragraph on page 394.

Zoom

It is now possible to zoom in the current form.

You can pass into "Zoom" mode by clicking on the magnifying glass or by clicking directly on the desired bar . The display percentages are 50%, 100%, 200%, 400% and 800%.

- When you click on the magnifying glass, the cursor takes the form of one . To increase the display percentage, click in the form. To reduce the display percentage, hold down **Shift** (the cursor turns into a) and click in the form.
- When you click on a percentage bar, the display is immediately modified.

In Zoom mode, all Form editor functions remain available.

Form Editor Toolbar

The toolbar of the Form editor offers a set of tools that can be used to manipulate and modify the form. Each window has its own toolbar:

The toolbar contains the following elements:

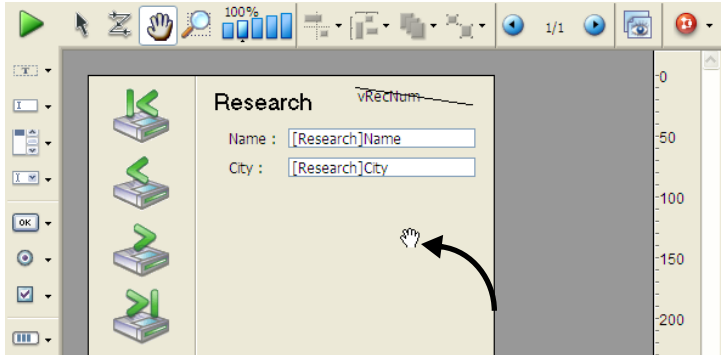
Execute the form: Used to test the execution of the form. When you click on this button, 4D opens a new window and displays the form in its context (list of records for a list form and current record page for a detail form). The form is executed in the main process.

Selection tool: Allows selecting, moving and resizing form objects. For more information, refer to the section "Managing Form Objects" on page 410.

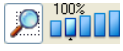
Note When an object of the Text or Group Box type is selected, pressing the **Enter** key lets you pass into editing mode.

Entry order: Switches to "Entry order" mode, where it is possible to view and change the current entry order of the form. For more information, refer to the "Data Entry Order" paragraph on page 475. Note that *shields* now allow viewing the current entry order, while still working in the form (see the "Using Shields" paragraph on page 434).

Moving: Passes into "Move" mode, where it is possible to reach any part of the form quickly by using drag and drop in the window. The cursor takes the shape of a hand:



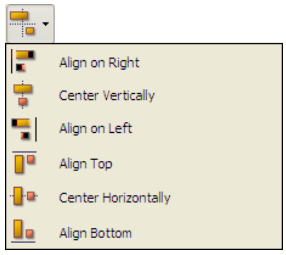
This navigation mode is particularly useful when zooming in the form.



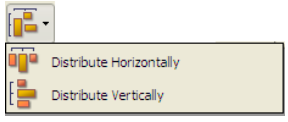
Zoom: Allows modifying the form display percentage (100% by default).

You can pass into “Zoom” mode by clicking on the magnifying glass or by clicking directly on the desired bar. This feature is detailed in the “Zoom” paragraph on page 376.

Alignment: This button is linked to a menu that allows aligning objects in the form. It is enabled (or not) depending on the objects selected.

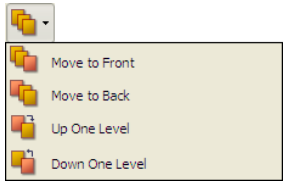


Distribution button: This button is linked to a menu that allows distributing objects in the form. It is enabled (or not) depending on the objects selected.



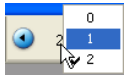
Note For more information about alignment and distribution of objects, refer to the section “Aligning Objects” on page 420.

Level button: This button is linked to a menu that allows changing the level of objects in the form. It is enabled (or not) depending on the objects selected. For more information, refer to “Layering Objects” on page 431.



Group/Ungroup: This button is linked to a menu that allows grouping and ungrouping selections of objects in the form. It is enabled (or not) depending on the objects selected. For more information, refer to "Grouping Objects" on page 419.

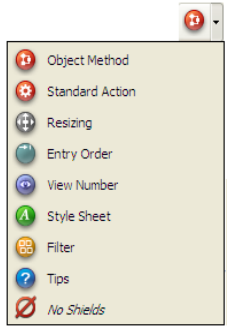
Display and page management: This area allows passing from one form page to another and adding pages. To navigate among form pages, click the arrow buttons, or click the central area and choose the page to display from the menu that appears:



If you click the right arrow button while the last form page is displayed, 4D allows you to add a page. For more information about this, refer to "Creating a Multi-page Form" on page 467.

Managing views: This button displays or hides the views palette. This new function is detailed in the "Using Object Views" paragraph on page 439.

Displaying shields: Each click on this button causes the successive display of each type of form shield. The button is also linked to a menu that allows directly selecting the type of shield to display:



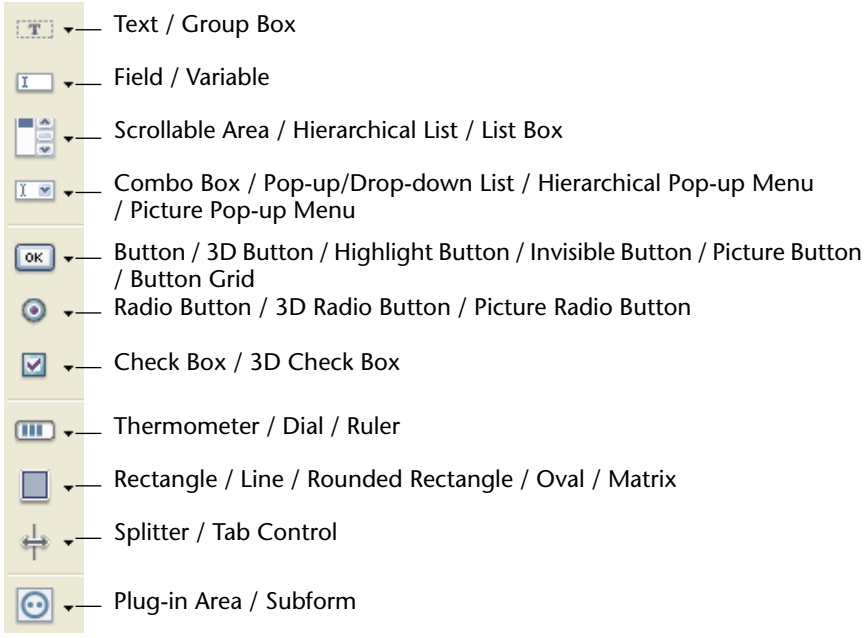
This function is detailed in the "Using Shields" paragraph on page 434.

User lock: When shown on the right side of the toolbar, this icon means that the form is "Editable by user." In this case, it cannot be edited directly; the form must first be unlocked by clicking this icon. For more information, please refer to the "Editable by User" paragraph on page 402.

Form Editor Object Bar

The object bar contains all the active and inactive objects that can be used in 4D forms. Some objects are grouped together by themes (buttons, radio buttons, menus, etc.). Each theme includes several alternatives that you can choose between.

Here are the object families available in 4D:



Using the Object Bar

The following describes how to use and select the objects of the object bar.

- To draw an object type, select the corresponding button and then trace the object in the form. After creating an object, you can modify its type using the Property List.

Note Hold down the **Shift** key as you draw to constrain the object to a regular shape. Lines are constrained to horizontal, 45°, or vertical, rectangles are constrained to squares, and ovals are constrained to circles.

- The current variant of the theme is the object that will be inserted in the form. When you click the right side of a button, you access the variant menu:

- When the object bar has the focus, you can select the buttons using the keys of the keyboard. The following associations are used:

Button	Selection key
Text / Group Box	T
Field / Variable	F
Scrollable Area / Hierarchical List / List Box	L
Combo Box / Pop-up/Drop-down List / Hierarchical Pop-up Menu / Picture Pop-up Menu	P
Button / 3D Button / Highlight Button / Invisible Button / Picture Button / Button Grid	B
Radio Button / 3D Radio Button / Picture Radio Button	R
Check Box / 3D Check Box	C
Thermometer / Dial / Ruler	I
Rectangle / Line / Rounded Rectangle / Oval / Matrix	S
Splitter / Tab Control	D
Plug-in Area / Subform	X

When a button is selected, you can scroll through its variants using the **Shift+Selection key** shortcut.
Help tips display the currently selected variant and the associated selection key of the object:

- You can click twice on the button so that it remains selected even after you have traced an object in the form (continual selection). This function makes creating several successive objects of the same type easier. To cancel a continual selection, click on another object or tool.

Description of Objects

Here is a brief description of the objects in the object bar:

- **Text**: Used to create or edit text on the form for labels, titles, instructions, and so on. For more information, see "Creating and Editing Text Areas" on page 446.
- **Group Box**: Used to create a box with a built-in label. This type of object is described in "Creating a New Form using the Wizard" on page 350.
- **Field**: Used to add a field to a form. For more information, see the section "Adding Fields to a Form" on page 485.
- **Variable**: Used to create text type variables. For more information, refer to "Enterable and Non-enterable Variables" on page 550.
- **Scrollable Area**, **Hierarchical List**, **List Box, Combo Box**, **Pop-up Menu**, **Drop-down List**, **Hierarchical Pop-up Menu**, **Picture Pop-up Menu**, **Buttons** and **Button Grid**, **Check Box**, **Radio Button**, **indicators**, **Splitter** and **Tab Control**: Used to create various types of active objects and their variants. For more information about this, refer to "Managing Active Objects on a Form" on page 537.
- **Graphic object tools**: Used to create graphic objects such as circles, squares, rectangles, rounded rectangles, and matrixes. For more information on how to create and use a matrix, refer to "Duplicating on a Matrix" on page 545.
- **Plug-in Area**: Used to create areas where plug-ins can be placed. Plug-ins can be used to bring additional functionalities to your 4D applications . For more information about plug-ins, please refer to "Installing Plug-ins" paragraph on page 47.
- **Subforms**: Used to create an area that displays multiple records from another table or a subtable. For more information, see the section "Adding a Subform to the Form" on page 621.

Form Editor Menus

When a Form editor window is the active window, the following menus are added to the menu bar:

- Form
- Object
- In addition to the standard menus, the Form editor also includes contextual menus that you can select at any time.

The Form Menu

Use the **Form** menu to organize form elements. Some of its commands display a hierarchical submenu. The figure below shows the **Form** menu.

The following is a list of the commands in the **Form** menu, with a brief description of their use:

- **Entry Order**: Used to create a custom entry order for data entry objects in an input form. When the entry order mode is selected, a check mark is displayed next to this menu command. For more information, see the section "Data Entry Order" on page 475.
- **Property List**: Used to display or hide the Property List window, which contains the properties of the form or its objects. A check mark appears next to the command when the Property List is displayed. For more information about the Property List, refer to "The Property List" paragraph on page 388.
- **Form Method**: Selecting this menu command opens the form's method in the Method editor. If no method is assigned to the form, a new blank method is opened in the Method editor. For more information, refer to chapter 8, "Methods" on page 757.
- **View List**: Used to display or hide the Views palette. A check mark appears next to the command when the palette is displayed. For more information about Views, refer to the "Using Object Views" paragraph on page 439.
- **Form Scaling**: Enlarges or reduces all form objects by a specified factor. For more information, see the section "Scaling a Form" on page 437.

- **Ruler Definition**: Used to set the scale of the form rulers. For more information, see the section "Using the Rulers" on page 418.
- **Magnetic Grid**: Used to turn on or off the grid to which objects can be aligned. When the grid is on, a check mark appears next to the command. To turn the grid off, select the command again. For more information, see the section "Aligning Objects" on page 420.
- **Display**: Used to display or hide Form editor elements (page 0, rulers, etc.) via a submenu. The commands of this submenu are described in the section "Showing/Hiding Elements in the Form Editor" on page 374.
- **Add Page**: Selecting this menu command adds a page after the last page.
- **Insert Page**: Selecting this menu command adds a page before the current page.
- **Delete Page**: Selecting this menu command deletes the current page.
- **Goto Page**: Selecting this menu command displays a hierarchical submenu that lets you choose between the form's existing pages. Selecting a page number automatically displays the corresponding page. This command is equivalent to the navigation pop-up menu located in the lower right corner of the editor window when the rulers are displayed. For more information about managing multi-page forms, refer to the section "Creating a Multi-page Form" on page 467

The Object Menu

Use the **Object** menu to modify and manipulate form objects. Some **Object** menu commands display a hierarchical menu of choices. The following illustration shows the **Object** menu.

Here is a list of the commands in the **Object** menu, with a brief description of their use:

- **Line Width**: Displays a hierarchical menu of line width choices for lines and borders.
- **Fill**: Displays a hierarchical menu of fill-pattern choices for objects.
- **Border**: Displays a hierarchical menu of border-pattern choices for objects.
- **Color**: Displays a hierarchical menu of color choices for objects.
- **Font**: Displays a hierarchical submenu used to apply a font to objects on the form and to specify a default font for text objects that you add to the form subsequently. The contents of the **Font** menu depend on the fonts installed on your system.

- **Style**: Displays a hierarchical submenu used to apply a text style, alignment, and font size to objects in the form, or to specify these attributes as defaults for objects that you add to the form subsequently.

Notes
- For more information about these commands, please refer to "Changing the Appearance of Objects" on page 443.
- All the graphic attributes of objects can also be set in the Property List.

- **Move to Front**: Used to move an object in front of all other objects.
- **Move to Back**: Used to move an object in back of all other objects.
- **Up One Level**: Used to move the object selection up one level towards the front.
- **Down One Level**: Used to move the object selection down one level towards the background.

Note For more information about level commands, please refer to "Layering Objects" on page 431.

- **Group**: Used to combine multiple objects in the form into groups that you can manipulate as a single object. For more information, see the section "Grouping Objects" on page 419.
- **Ungroup**: Used to separate grouped objects into individual objects. For more information, see the section "Grouping Objects" on page 419.
- **Align**: Selecting this menu command displays a submenu that allows you to choose between several alignment and distribution types. By selecting the Alignment submenu item, you can also display the alignment assistant. For more information, refer to "Aligning Objects" on page 420.
- **Duplicate on Matrix**: Selecting this menu command duplicates an object X-1 times, using a matrix object that contains X cells. This menu command is enabled only if an object and a matrix are selected and the object is contained in the upper left corner cell of the matrix. For more information, refer to "Duplicating on a Matrix" on page 545.

- **Duplicate Many**: Selecting this menu command displays the Duplicate Many dialog box. This dialog box allows you to duplicate the same object several times and automatically assign each instance a numbered name. For more information, refer to "Duplicating Objects" on page 427.
- **Show Format**, **Show Resource**, **Show Name**: Controls what is displayed on the form for objects whose text is actually stored in resources. Show Resource shows the STR# or XLIFF resource in which the text of the object is stored. These menu commands also allow you to display the references for field and table labels if they have been inserted using dynamic references. For more information, refer to the sections "Working with Static Text Areas" on page 445 and "Inserting Dynamic Table and Field Names" on page 487 as well as to appendix C "XLIFF Architecture"on page 1293.
- **Object Method**: Selecting this menu command opens the object's method in the Method editor. If there is no object method assigned to the object, a new blank method is opened in the Method editor. For more information on this point, please refer to Chapter 11 "Methods" on page 757.
- **Clear Object Method**: Used to remove an object method from a selected object. To do so, select the object and choose this command from the menu.
- **Insert an OLE Object**: This command allows you to create a plug-in area that is designed to become an OLE area. The use of OLE areas is described separately. For more information on this point, please refer to the *4D OLE Tools* manual.

Contextual Menus

You can use contextual menus in the Structure editor. To use a contextual menu, click on an object or area with the **right button** of the mouse.

With the exception of the **Property List** and **View List** commands (used to display/hide the corresponding palettes) and standard menu commands such as **Copy** and **Paste**, commands in contextual menus vary according to the current editor and the object on which you clicked:

- If you clicked on a form object, the contents of the contextual menu are object-related: **Object Method**, **Color**, **Align**, **Border Line Style**, **Automatic Size**, **Standard Action** (depending on the object) and **Level**.

The **Select Similar Objects** command is used to select all objects of the same type as the object that is clicked (see the "Selecting Objects" paragraph on page 410).
The **Use as Template** command lets you use the characteristics of the object as a template for the creation of all other objects of the same type (see the "Setting an Object Template" paragraph on page 430).
Moreover, depending on the object type, additional commands are provided:

- **Attached Field** if the object is a field,
- **List** if the object is a hierarchical list,
- **Edit Picture...** if the object uses a picture (displays the picture in the picture library),
- **Edit Form...** if the object is a subform,
- **Default Values...** if the object is a scrollable area, a combo box, a pop-up/drop-down list or a tab control.

- If you click outside any object, the contextual menu's items apply to the form: **Form Method**, **Turn Magnetic Grid On** and **Display** (see the "Showing/Hiding Elements in the Form Editor" paragraph on page 374).

The Property List

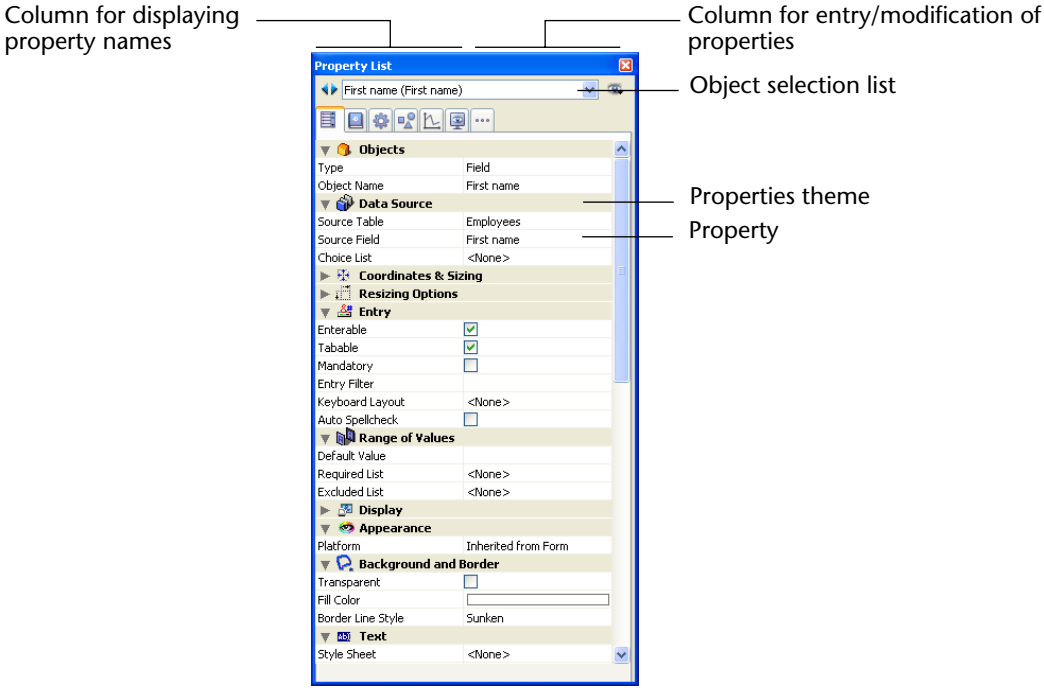
Both forms and form objects have properties that control access to the form, the appearance of the form, and the behavior of the form when it is used. Form properties include, for example, the form's name, its menu bar, and its size. Object Properties include, for example, an object's name, its dimensions, its background color, and its font.

This section describes how to display and modify form and object properties using the Property List. For more information on form properties, refer to "Setting Form Properties" on page 394. For more information on object properties, refer to "Managing Form Objects" on page 410 and to chapter 6, "Fields and Active Objects in Forms" on page 483.

▶ To display/hide the Property List:

1 **Choose Property List from the Form menu or from the contextual menu of the Form editor (right-click in the editor window outside any object).**
OR
Use the Ctrl+Shift+Space bar (Windows) or Command+Shift+Space bar (Mac OS) shortcut.

The Property List appears:



Selection of Contents

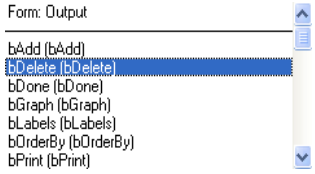
The Property List displays either the properties of the form or those of the object(s) selected:

- When no object is selected, the Property List displays the form properties,
- When an object or set of objects is selected, the Property List displays the properties of the object or the common properties of all the objects selected.

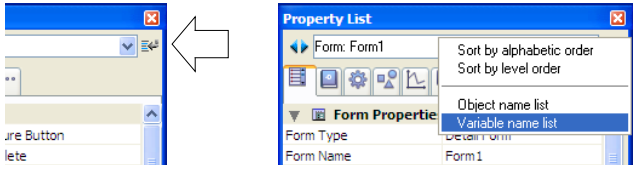
  You can select the elements whose properties you want to display by clicking on them in the form or using the selection list at the top of the Property List window.

  The selection list contains three different parts, separated by a line:
  - Form name,
  - Dynamic objects (linked to a variable),

- Static objects (not linked to a variable).



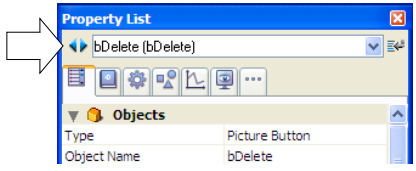
For each dynamic object, the list displays either the name of the variable followed by the object name between parentheses, or the object name only. You can set this display using the menu associated with the list:



The command indicates the current type of display.

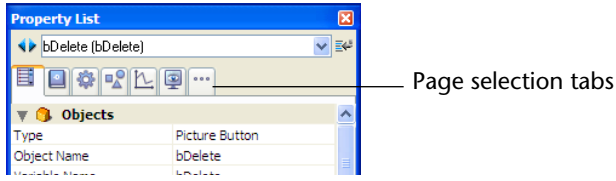
You can also modify the sort of each part of the list. By default, the list is sorted according to level order (from foreground toward the background). The associated menu allows you to sort the list by alphabetical order.

The buttons located to the left of the list allow you to select each object in the list successively, toward the top or bottom:



Property Display Pages

You can choose how information in the Property List is displayed using the page selection tabs:



- The first page (Main) displays all information related to the selected object using different themes (Data Source, Coordinates & Sizing, Events, etc.). You can expand or collapse each theme by clicking on the expansion icon to the left of its name; this lets you display only the properties with which you want to work.
  The display is contextual: only themes and properties relevant to the form/type of object selected are displayed. The display of certain properties is also dynamically modified depending on the value of other properties. If, for instance, you set the Enterable property for a field, the Tabable property is then displayed.

- The six other pages each contain specific information related to the selected object. In addition, all properties of each theme are displayed, regardless of the object type. Non-pertinent properties are dimmed. This alternative provides a more global vision of the settings associated with a given object. The following describes the six theme pages:
  - Settings: Contains properties that define object identity (type, name, data source, etc.).
  - Behavior: Contains properties that set the dynamic behavior of the objects (associated method, drag and drop, form events).
  - Dimensions: Contains properties that set coordinates, size and object resizing.
  - Value: Contains properties that define the editing and display of enterable objects (enterable, shortcut, display format, etc.).
  - Appearance: Contains properties that set the graphic appearance of objects.
  - Specific: Contains properties of the selected object type (tips for fields, animated picture buttons, print settings for forms, etc.). Unlike other theme pages, non-relevant properties for the object do not appear.

Navigation Shortcuts

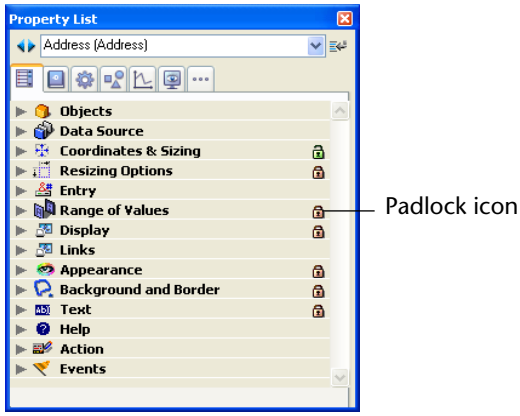
You can navigate in the Property List using the following shortcuts:

- **Arrow keys** ↑↓: Used to go from one cell to another.
- **Arrow keys** ←→: Used to expand/collapse themes or menus.
- **PgUp** and **PgDn**: Used to select the first or last visible cell of the Property List.
- **Home** and **End**: Used to select the first or last cell of the Property List.

- **Ctrl+click** (Windows) or **Command+click** (Mac OS) on an event: Used to select/deselect every event in the list, according to the initial state of the event on which you clicked.
- **Ctrl+click** (Windows) or **Command+click** (Mac OS) on a theme label: Used to Collapse/Expand every theme in the list.

Padlock Icons

When the **Editable by user** property is applied to the form, padlocks are displayed in the Property List indicating the properties that are modifiable (or not) by the user:



For more information, refer to the "Editable by User" paragraph on page 402.

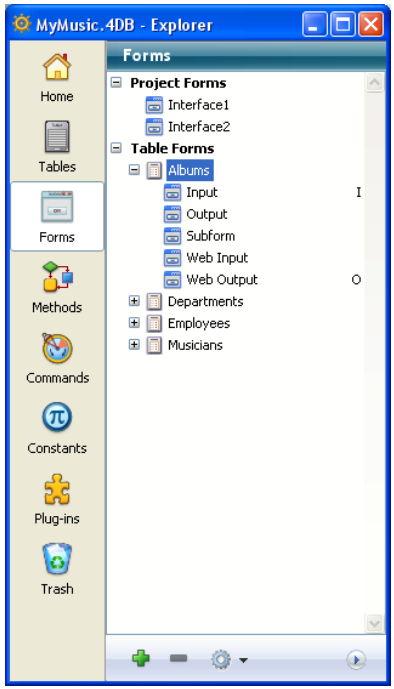
Opening a Form in the Form Editor

You can edit an existing form in the Form editor at any time.

► To open a form in the Form editor:

1 **Choose Open > Form... from the File menu.**
OR
Hold down the Ctrl key (Windows) or the Command key (Mac OS) and double-click the table title whose forms you want to open in the Structure editor.
OR
In the Structure editor, click the table name using the right mouse button, then select Show Forms... from the contextual menu.

4D displays the Forms page of the Explorer. You can expand any of the table names to display the forms associated with it.



If you have double-clicked a table's name or used the contextual menu of the Structure editor, that table is already selected.

2 **If necessary, expand the table name that contains the form you want to modify.**

3 **Click with the right mouse button on the name of the form to be modified, then select Edit Form... in the contextual menu.**
OR
Select the form then choose Edit Form... in the options menu of the Explorer.
OR
Double-click the name of the form or in the preview area.
4D displays the form in a Form editor window.

Note These actions are also available on the Home page of the Explorer.

4D Server Object locking occurs when two or more users attempt to modify the same form simultaneously. If a user is modifying a form in the Design environment, the form is locked. Other users cannot modify that same form until the first user frees the form by closing it.

Setting Form Properties

Form properties are set in the Property List.

Note Certain form properties can be set using the Form Properties window specific to the Explorer. For more information about this point, refer to the "Setting Form Properties via the Explorer" paragraph on page 115.

Renaming the Form

You can rename the form in the Property List or using the Forms page of the Explorer.

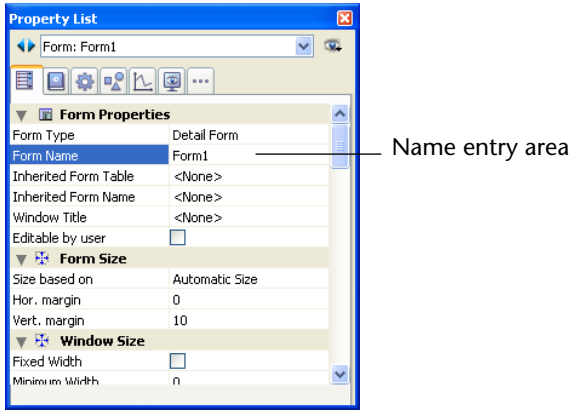
You use the names of forms when you are establishing default input and output forms for a table and in commands that accept a form name as a parameter, such as INPUT FORM and OUTPUT FORM.

You cannot use the same name for more than one project form, or for more than one form per table. This will confuse 4D when you try to refer to a form by name. You can, however, use the same form name with different tables. For example, you can name all your input forms "Input" and all your output forms "Output."

► To rename a form using the Property List window:

1 **Display the form properties in the Property List.**

2 **Enter a name in the Form Name area.**



Note You can also rename a form using the Form Properties window specific to the Explorer (see the "Setting Form Properties via the Explorer" paragraph on page 115).

► To rename a form using the Explorer:

1 **Click the Forms button in the Explorer.**
A hierarchical list of tables and forms appears.

2 **Hold down the Alt key (Windows) or the Option key (Mac OS) and click on the form name.**
OR
Click twice on the name of the form you want to change.
The form name becomes editable.

3 **Enter the new name.**

4 **Press Tab or click anywhere outside the entry area to save the new name.**

If you rename a form that is referred to elsewhere in the database (such as in methods), be sure to update the references to the form. To do this, you can use the search and replace function in the Design environment (see the "Searching and Replacing in the Design" paragraph on page 140).

Form Access

You can control access to a form by setting Access and Owner privileges for groups of users. A single group can be assigned for each privilege using the Access and Owner drop-down lists. For information about creating a password access system with users and groups, see the chapter "Users and Groups" on page 831.

The Access drop-down list controls which group can use the form in the records display window or in custom applications. If a user that is not in this group attempts to use the form, 4D displays a message saying that the user's password does not allow him or her to use the form.

The Owner drop-down list controls which group can edit the form in the Design environment. If a user who is not in this group attempts to open the form in the Design environment, 4D displays a message saying that the user does not have the access privileges needed to edit the form.

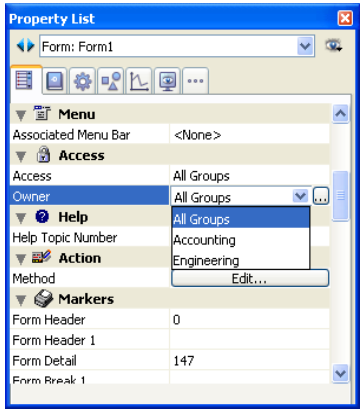
Users who are assigned to both groups can use the form in both the Design environment and in custom applications.

► To set access privileges for a form:

1 **Display the form properties in the Property List.**

2 Use the Access and Owner drop-down lists to make the desired access privilege assignments.

The names of existing groups are displayed in each drop-down list.



The [...] button can be used to access the Groups editor directly in the Tool Box of 4D.

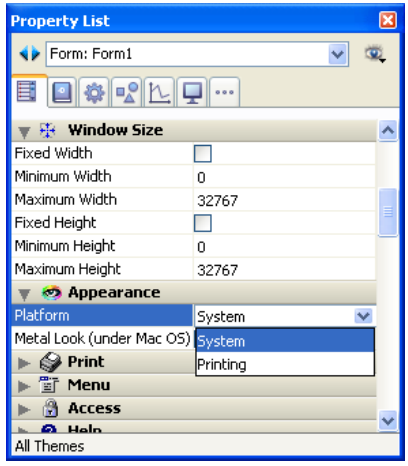
Note You can also assign access groups to a form using the Form Properties window specific to the Explorer (see the "Setting Form Properties via the Explorer" paragraph on page 115).

Platform

The platform interface property is used to set the appearance of a form according to the context of its execution.

This property can also be set individually for each form object (see the "Platform" paragraph on page 444).

Two interface properties are available: **System** and **Printing**. They are used as follows: When displayed on screen, a form must respect the current operating system interface (**System**). When it is set to be printed, the appearance of objects must be adapted (**Printing**), regardless of the platform.



Note Additional options are available in converted databases, see the "Compatibility" paragraph on page 398.

System

This property allows adapting the appearance of the form or object automatically depending on the current platform on which 4D is running.

- When the form is displayed under Mac OS X, the form or object has a Mac OS X look and feel,
- When the form is displayed under Windows, the form or object has the look and feel based on the current "Appearance" setting of the control panel.

This mode is used by default for all the forms and objects in databases created with 4D version 2004 or higher.

Printing

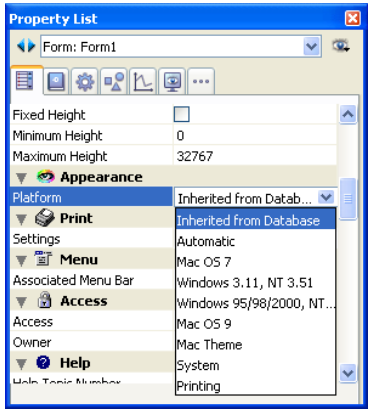
This property allows adapting the appearance of the form or object for printing: any object or graphic item (button, check box, tab, line, etc.) is drawn as a vector in order to produce a satisfactory result for printing.

The same result is obtained regardless of the platform on which the form is displayed or printed.

Compatibility

Different properties are available depending on the origin of the open databases (created or converted).

- In databases created with 4D version 2004 or higher, only **System** and **Printing** platform properties are available (as well as the **Inherited from Form** property for objects). They replace all other options available in earlier versions of 4D.
- In databases that were converted from earlier versions of 4D, these properties are added to other interface options, which are kept for compatibility reasons. The former platform properties can still be set in the database Preferences (see the "Platform" paragraph on page 178), or at the level of forms and objects using the Property List.

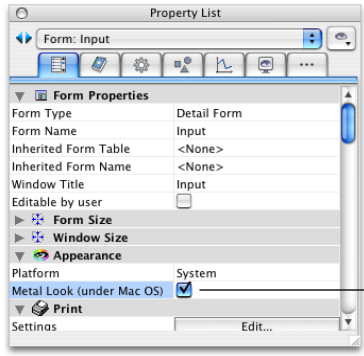


The use of these commands is not recommended because they do not allow you to take advantage of new interface features of 4D. Generally speaking, you should use the **System** and/or **Printing** properties to make sure that databases will be compatible with future releases of 4D.

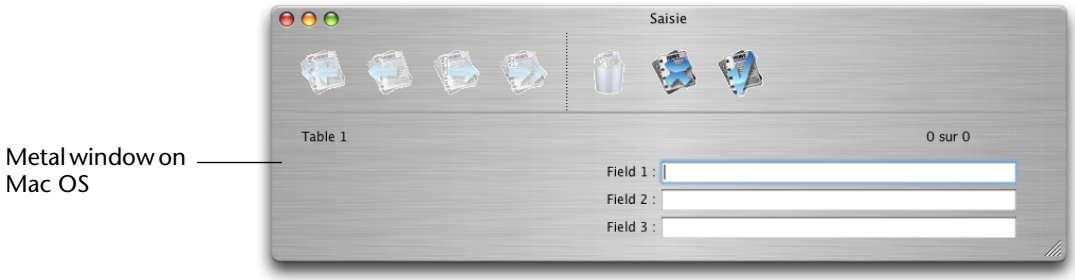
Note You can also set the platform of a form using the Form Properties window specific to the Explorer (see the "Setting Form Properties via the Explorer" paragraph on page 115).

Metal Look Under Mac OS

The form property **Metal Look (under Mac OS)** in the "Appearance" theme can be used to activate the metal look for the form when it is displayed using the Open form window command on Mac OS:



This look is found throughout the Mac OS X interface:



Note Under Windows, this property has no effect.

If the form is displayed in a window that was not created by Open form window (for example in the Design mode), the property will not be taken into account.

Under Mac OS, the metal look is previewed in the Form editor when the **Metal Look** option is checked and when the form limits are displayed.

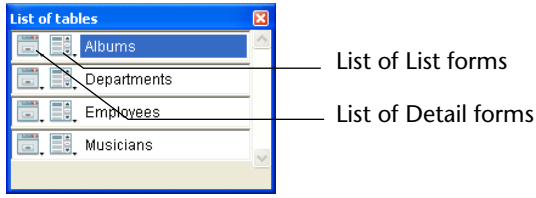
Form Type

You can change the form type, i.e. its destination. For project forms, there are two types available: **Detail Form** and **Detail Form for Printing**. For table forms, two additional types are also available: **List Form** and **List Form for Printing**.

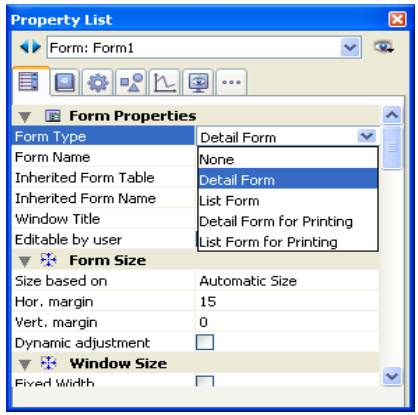
This property determines the options that appear in the Property List for the form.

It also allows you to restrict the number of forms displayed in the current Input and Output form selection lists (the **List of tables** window).

Only forms whose type corresponds to the list are displayed.



The Form Type property is found at the top of the Property List:

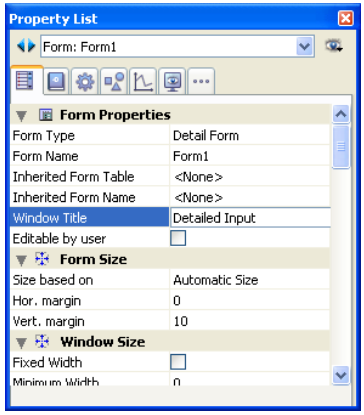


When the form type is **None**, it is displayed in both menus of the List of tables window.

Note You can also set the form type using the Form Properties window specific to the Explorer (see the "Setting Form Properties via the Explorer" paragraph on page 115).

Default Window Title

The default window title is used when the form is opened using the Open window and Open form window functions in the Application environment. The default window title appears in the Title bar of the window. To set the default window title, enter it in the Window Title entry area.



You can use dynamic references to set the window title. The reference is resolved when the INPUT FORM command is called (if the * parameter is passed and if it is followed by a call to Open window) and when the Open form window command is called.

The following types of dynamic references can be inserted in a window's title:

- An **STR# or XLIFF resource reference**: The syntax to apply is ":16000,2" where 16000 is the resource number and 2 is its element. In the case of an XLIFF reference, 16000 designates the **id** attribute of the **group** element and 2 designates the **id** attribute of the **trans-unit** element.
- A **standard XLIFF reference**: The syntax to apply is of the ":xliff:WindowName" type. The referenced value (*WindowName*) designates the **resname** attribute of the **trans-unit** element.

Note For more information about XLIFF references, please refer to appendix C "XLIFF Architecture"on page 1293.

- A **table or field label**: The syntax to apply is <?[TableNum]FieldNum> or <?[TableName]FieldName>. For more information, refer to "Inserting Dynamic Table and Field Names" on page 487.

- A **variable or a field**: The syntax to apply is <VariableName> or <[TableName]FieldName>. The current value of the field or variable will be displayed in the window title.

Notes

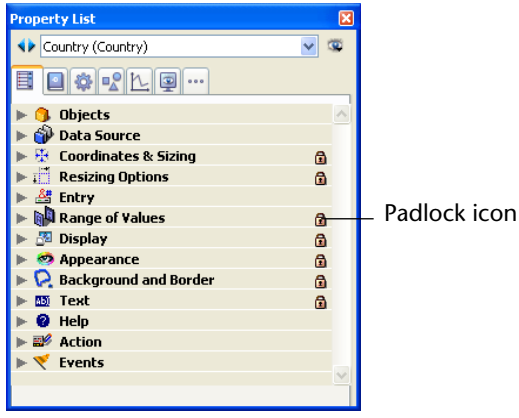
- The number of characters for a window title is limited to 31.
- You can also set the window title using the Form Properties window specific to the Explorer (see the "Setting Form Properties via the Explorer" paragraph on page 115).

Editable by User

The **Editable by user** property is used to indicate that the form can be customized by the users via the user form editor in the Application mode. This editor is described in the chapter "User Forms" on page 685. Putting this mechanism into place requires additional settings as well, which are described in the "User Forms" section of the *4D Language Reference* manual.

When the **Editable by user** property is checked, the form is locked. When you open a form with this property, a padlock icon appears in the toolbar and it is not possible to make any modifications. You must click on this icon to unlock the form — in this case, any user forms become obsolete and must be regenerated.

When the **Editable by user** property is checked, the Property List also has padlocks indicating the locked properties.



You can click on these icons to "open" the padlocks and thus make the corresponding properties editable in the user form editor.

 Open padlock

For more information about the user form editor, refer to the chapter "User Forms" on page 685.

Note You can also set the **Editable by user** property using the Form Properties window specific to the Explorer (see the "Setting Form Properties via the Explorer" paragraph on page 115).

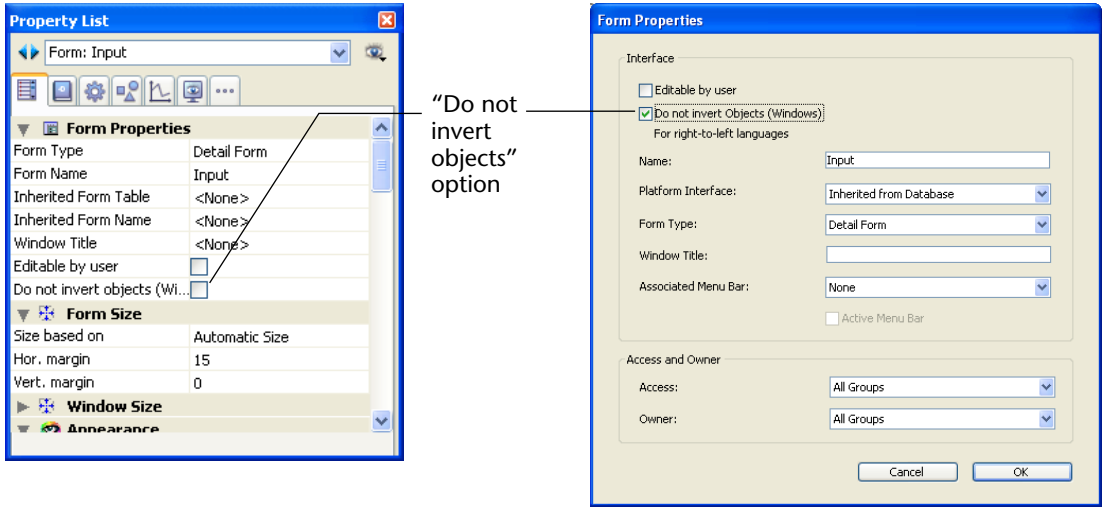
Do Not Invert Objects (Windows)

The **Do not invert objects** option can be used to disable the "right-to-left" mode for the form.

The "right-to-left" mode (Windows only) causes the database form objects and titles to be inverted. It is used in the context of languages that are read from right to left (such as Arabic). For more information about activating this mode, please refer to the "Right-to-left Languages" paragraph on page 201.

When the **Do not invert objects** option is checked for a form, the right-to-left mode is never applied to this form in Application mode, regardless of the database configuration. This option can be used, for example, to set "international" forms within a database configured in right-to-left mode.

This option is found in the Property List, as well as in the "Form Properties" dialog box that can be accessed from the 4D Explorer:



Note This property can also be managed by programming using the GET FORM PARAMETER command.

Assigning a Menu Bar to a Form

When you create an application, you can create custom menus. Custom menus allow you to add menu commands for automating specific tasks in the database, such as, for example, creating a report.

Custom menus are created in the Menu Bar editor. Each menu bar that you create includes at least one menu and is assigned a unique ID number and name. For more information on creating menu bars, menus and menu commands, refer to the chapter “Custom Menus” on page 859.

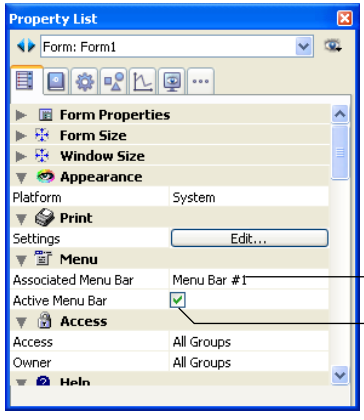
In the Application environment, a menu bar that is assigned to a form is added to the right of the current menu bar. The menu bar added disables other menus unless the **Active Menu Bar** option is selected. If the menu bar of the form is identical to the current menu bar, it is not added. The form menu bar will operate for both input and output forms.

► To assign a menu bar to a form:

1 **Display the form properties in the Property List.**

2 **Select a menu bar from the “Associated Menu Bar” List.**

The “Active Menu Bar” option then appears.



3 **If you want to use this form in a custom application, select the “Active Menu Bar” option.**

This option tells 4D not to disable the current menu bar. If this option is not selected, 4D disables the current menu bar and only permits access to the form's menu bar.

Note You can also associate a menu bar with a form using the Form Properties window specific to the Explorer (see the "Setting Form Properties via the Explorer" paragraph on page 115).

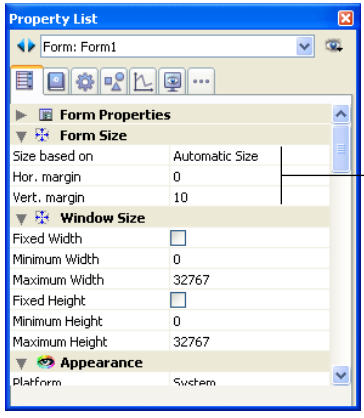
Form and Window Size

A form is always displayed in a window. 4D lets you set the size of both the form and the window, as well as their respective behavior when resized.

These properties are interdependent and your application interface results from their interaction. Keep in mind that the result also depends on the resizing properties assigned to each form object (for more on this subject, refer to the "Setting Resizing Properties" paragraph on page 416).

Form Size

You set the form size properties using the "Form Size" theme of the Property List.



The following choices are available:

- **Size based on: Automatic Size**: The size of the form will be that necessary to display all the objects, to which will be added the margin values entered in the **Hor. margin** and **Vert. margin** fields (in pixels).
- **Size based on: Set size**: The size of the form will be based on what you enter (in pixels) in the **Width** and **Height** fields (when you select the Set size option, the Hor. margin and Vert. margin fields change to Width and Height).

- **Size based on: object**: The size of the form will be based on the position of the selected form object. For example, if you choose an object that is placed in the bottom-right part of the area to be displayed, the form size will consist of a rectangle whose upper left corner will be the origin of the form and the lower right corner will correspond to that of the selected object, plus any margin values.
  You can choose this option when you want to use active objects placed in an offscreen area (i.e., outside the bounding rectangle of the window) with an automatic size window. Thanks to this option, the presence of these objects will not modify the size of the window.

- When you select the **Automatic Size** option or a size based on an object, you have the **Hor. margin** and **Vert. margin** fields. You can then enter values (in pixels) to set additional margins to be added to the edges of the form.

  These values also determine the top and right-hand margins of forms used in the Label editor (see the "Printing Labels" paragraph on page 680).

Note For output forms, only the **Hor. margin** or **Width** fields are available.

- **Dynamic adjustment**: The **Dynamic adjustment** property is available in the "Form Size" theme for forms converted from a previous version of 4D, when the **Automatic Size** property is selected. This property is used to modify the resizing mode of forms.
  In previous versions of 4D, when a form had the **Automatic Size** property, the form size was calculated only at the moment when the form was opened. If any modifications were made subsequently using commands such as MOVE OBJECT, the size of the form was not adjusted. Starting with 4D 2004, the form size is dynamically adjusted in this case. This principle is activated for new forms. On the other hand, for compatibility reasons, the forms of converted databases do not benefit from this property by default. If you want the size of converted forms to be dynamically adjusted, you can check the **Dynamic adjustment** option.

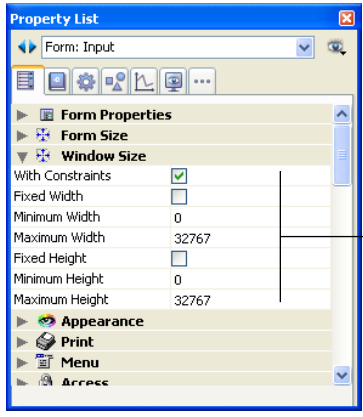
Window Size

When an input form is displayed in a custom application, you ordinarily open the form using the Open window or Open form window functions.

Open window lets you specify the top, left, bottom, and right coordinates of the window as well as the window type. In this case, the size of the window does not depend on that of the form. On the other hand, the resizing possibilities will depend on the options set in the “Form Size” theme and on the window type.

Open form window creates a new window based on the sizing and resizing properties of the form passed as parameter.

You can set form window resizing in the Property List:



The following options are available:

- **With Constraints**: This property is available for forms converted from previous versions of 4D. It is used to reproduce the behavior of the prior “Resizable” form property: when this option is not checked, the mechanisms that handle object resizing and window size constraints are disabled. This way, the user can freely resize the form window but the objects it contains are neither resized nor moved. The minimum/maximum or fixed size properties as well as the resizing properties of the objects are ignored.
  This corresponds to the behavior of former versions of 4D and must only be used for compatibility reasons within specific interfaces. The behavior of 4D databases created with version 2004 and higher corresponds to the checked option (standard mode).
- **Fixed Width**: If you check this option, the window width will be locked and it will not be possible for the user to resize it.
  If this option is not checked, the width of the form window can be modified. In this case, the **Minimum Width** and **Maximum Width** entry areas can be used to determine the resizing limits.

- **Fixed Height**: If you check this option, the window height will be locked and it will not be possible for the user to resize it.
If this option is not checked, the height of the form window can be modified. In this case, the **Minimum Height** and **Maximum Height** entry areas can be used to determine the resizing limits.

  As a general rule, it is necessary to prevent the user from hiding enterable areas and control buttons.

Form Events

You can write a form method using the Method editor. If you do so, you should indicate which form events should be executed. This is carried out using the Events theme of the Property List. When the form is used, only the events that you select will actually occur.

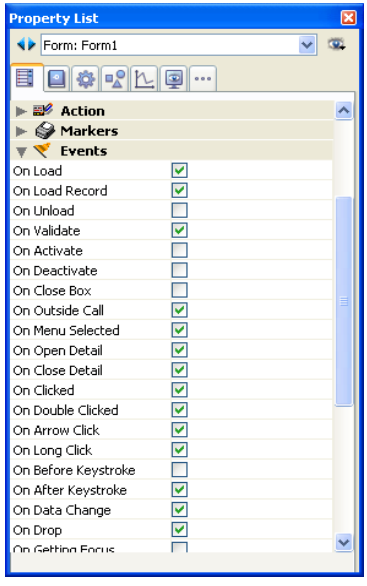
For information on the available events, see the Form event command in the 4D *Language Reference* manual.

Your database will run faster if you deselect superfluous events.

► To activate events for the form:

1 **Display the form properties in the Property List.**

2 **Expand the "Events" theme.**
The list of events is then displayed:



A check box next to the event indicates that the event will be handled when the form is used.

3 **Select only the events that are needed.**

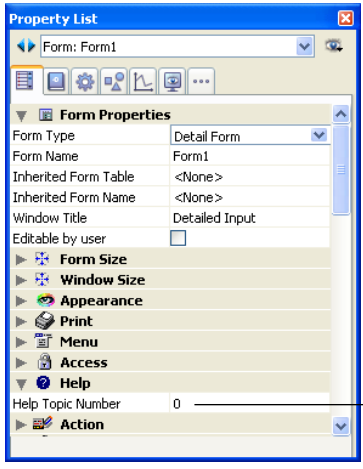
To select or deselect all events, hold down **Ctrl** (under Windows) or **Command** (under Mac OS) and click an event.

Contextual On-line Help

4D allows you to associate a custom on-line help file with each database. The creation of help files is described in appendix A "Assigning a Custom Help File"on page 1285.

Help files can be contextual, which means that they can display information related to the context from which they were called. To do so, you can associate a precise section of this help file with each of the database's forms.

This is done in the "Help" theme of the Property List:



You can repeat this procedure for each form that needs to be assigned a help file.

Make sure you assign help topic numbers that match numbers defined in the help file. For more information on this point, refer to appendix A "Assigning a Custom Help File"on page 1285.

Managing Form Objects

You customize a form by creating and manipulating objects in the form. You use the Form editor to draw the objects, modify them, arrange them, set their properties, or delete them.
You can select and modify any object in a form, including the fields, buttons, and graphic objects created by the Form Wizard.

Selecting Objects

Before you can perform any operation on an object (such as changing a line width or font), you need to select the object that you want to modify.

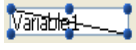
When the Form Wizard creates a form, it creates one or more display pages and a background page. The Form Wizard may place buttons, the form title, and decorative rectangles on the background page. Objects on the background page are selectable only from the background page. If you have difficulty selecting an object that was created by the Form Wizard, switch to the background page by clicking on the button or by choosing **0** in the pages menu and try again.

You can also choose to hide any element located on the background page by deselecting **Page 0** from the **Display** submenu of the **Form** menu (or the Form editor contextual menu).

► To select an object using the toolbar:

1 **Click the Arrow tool in the toolbar.**
When you move the pointer into the form area, it becomes a standard arrow-shaped pointer.

2 **Click the object you want to select.**
Resizing handles identify the selected object.



► To select an object using the Property List:

■ **Choose the object's name from the Object List drop-down list located at the top of the Property List.**
Using these two methods, you can select an object that is hidden by other objects or located outside the visible area of the current window.

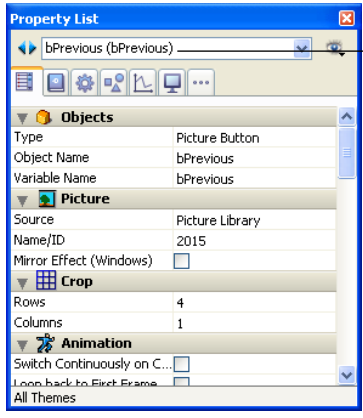
To deselect an object, click outside the object's boundary or **Shift+click** the object.

Note It is also possible to select objects by double-clicking them in the result window of an overall search in the database. For more information on this point, refer to the "Results Window" paragraph on page 146.

Displaying or Modifying Object Properties

Object properties can be displayed and modified using the Property List. To display the Property List, you can use the **Command+Shift+Space bar** (Mac OS) or **Ctrl+Shift+Space bar** (Windows) shortcut, or double-click on the object, or choose **Property List** from the **Form** menu (or the contextual menu of the editor).

Once this palette is displayed, select the object from the drop-down list or click on the object concerned directly in the editor window.



Viewing or Modifying an Object's Method

To view, modify or create an object method, hold down the **Alt** key (under Windows) or **Option** key (under Mac OS) and click on the object[1]. You can also click on the object and select the **Object Method** command in the **Object** menu (or in the contextual menu of the editor). It is also possible to click on the **Edit...** button for the Method line of the object found in the "Action" theme of the Property List.

For more information about object methods, refer to the sections "Using Object Methods with Fields and Objects" on page 616 and "Where to Put an Object Method" on page 765.

1. Only active objects can have methods, but all objects have properties. If an object cannot have a method, the Property List contains no Object Method line.

Selecting Multiple Objects

You may want to perform the same operation on more than one form object — for example, to move the objects, align them, or change their appearance. 4D lets you select several objects at the same time.

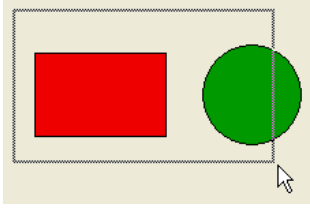
There are several ways to select multiple objects:

- Choose **Select All** from the **Edit** menu to select all the objects.
- Click with the **right button** of the mouse on the object and choose the **Select Similar Objects** command in the contextual menu.
- Hold down the **Shift** key and click the objects you want to select.
- Start at a location outside the group of objects you want to select and drag a marquee (sometimes called a selection rectangle) around the objects.
- Hold down the **Alt** key (Windows) or the **Option** key (Mac OS) and draw a marquee. Any object that is completely enclosed by the marquee is selected.

  You draw a marquee with the arrow pointer. A marquee defines a rectangular region that selects objects it surrounds or touches.

  To select objects by drawing a marquee around them, you must press the mouse button down and start dragging *in an area that contains no objects*. When you release the mouse button, if any part of an object lies within the boundaries of the selection rectangle, that object is selected.

  The figure below shows a marquee being drawn to select two objects:



To deselect an object that is part of a set of selected objects, hold down the **Shift** key and click the object. The other objects remain selected. To deselect all the selected objects, click outside the boundaries of all the objects.

Moving Objects

You can move any graphic or active object in the form including fields and objects created with a template.

When moving an object, you have the following options:

- Move the object by dragging it.
- Move the object one pixel at a time using the arrow keys.
- Move the object by steps using the arrow keys (20-pixel steps by default).
- Manually enter the coordinates of the object in the Property List (this point is described in "Resizing Objects" on page 414).

▶ To move an object by dragging:

1 **Select the object or multiple objects that you want to move.**

2 **Move the pointer over the selected object or one of the objects in a selected set of objects and drag to the new location.**

4D displays markers that show the location of the object's boundaries in the rulers so that you can place the object exactly where you want it. As you begin dragging the selected object, its handles disappear.

Be careful not to drag a handle. Dragging a handle resizes the object.

Note You can press the **Shift** key to carry out the move with a constraint.

When the magnetic grid is on, objects are moved in stages indicating noticeable locations. For more information about this point, refer to the "Using the Magnetic Grid" paragraph on page 423.

3 **Release the mouse button to complete the move.**

▶ To move an object one pixel at a time:

1 **Select the object or objects you want to move.**

2 **Use the arrow keys on the keyboard to move the object.**

Each time you press an arrow key, the object moves one pixel in the direction of the arrow.

▶ To move an object by steps:

1 **Select the object or objects you want to move.**

2 **Hold down the Shift key and use the arrow keys to move the object.**

By default, steps are 20 pixels at a time. You can change this value on the Form Editor page of the Preferences (see the "Move" paragraph on page 181).

Resizing Objects

You can change the size of any object that appears on the form. 4D lets you stretch or shrink objects on the form.

When resizing objects, you have the following four options:

- Resize an object by dragging a resizing handle.
- Resize an object one pixel at a time by using the **Ctrl** key (Windows) or the **Command** key (Mac OS) and the arrow keys.
- Resize an object by steps (by default, 20 pixels at a time).
- Manually enter the dimensions of the object in the Property List.

▶ To resize an object by dragging:

1 **Select the object you want to resize.**

2 **Move the pointer over one of the four handles that appear on the selected object.**

The pointer changes into a multi-directional arrow ✥ and the handles disappear.

3 **Drag the handle toward the center of the object to shrink it.**
OR
Drag the handle away from the object's center to enlarge it.

4D resizes the object.

As you drag the handle, the corner of the object opposite the dragging handle remains stationary.

Notes
• If you press **Shift** and then drag the handle, the movement is constrained. Lines can then be only vertical, 45°, or horizontal, rectangles can only be square, and ovals can only be circular.
• If the magnetic grid is on, manual resizing will be carried out in 5-pixel increments. For more information about the magnetic grid, refer to the "Using the Magnetic Grid" paragraph on page 423.

▶ To resize an object one pixel at a time:

1 **Select the object you want to resize.**

2 **Hold down the Ctrl key (Windows) or Command key (Mac OS) and use the arrow keys to resize the object.**

Pressing the up or down arrow keys resizes the object's height while pressing the left or right arrow keys resizes the object's width.

► To resize an object in steps:

1 **Select the objects you want to resize.**

2 **Hold down the Shift+Ctrl keys (Shift+Command keys under Mac OS) and use the arrow keys to resize the objects.**

By default, steps are 20 pixels at a time. You can change this value on the Form Editor page of the Preferences (see the "Move" paragraph on page 181).

► To resize an object by entering coordinates:

1 **Display the Property List and click the object you want to resize.**

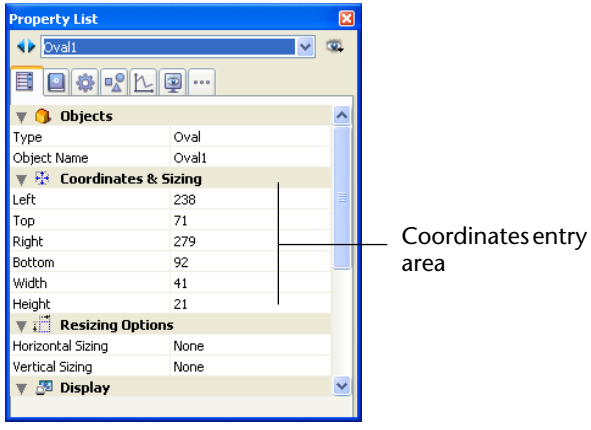
OR

Double-click the object.

2 **In the Property List, expand the "Coordinates & Sizing" theme.**

The coordinates of the object appear. They are displayed in pixels, centimeters, or inches (depending on the ruler units).

The upper-left coordinates of the form area are 0,0.



3 **Enter new values in the coordinate entry areas.**

4D moves the boundaries of the object to the positions you entered. Depending on the values you use, the object may be resized or moved (or both).

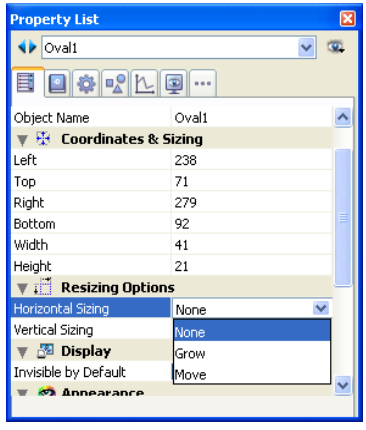
Setting Resizing Properties

In Application mode, when a user resizes the window displaying a form, the objects it contains are either resized or moved.

Automatic resizing works when a user resizes a window that displays a form. Automatic resizing causes an object to grow as the form is enlarged (or become smaller as the window is reduced). For example, if you use a rectangle that encloses the fields on an entry form, automatic resizing causes the rectangle to grow to the edges of the window as the user enlarges the window.

You can also enable automatic repositioning. Automatic repositioning moves an object either horizontally or vertically as the form is resized. When automatic repositioning is on, 4D tries to keep the object in view as the user reduces the size of the window. For example, if the user resizes a row of buttons so that some of the buttons become obscured, automatic repositioning tries to move the buttons either horizontally or vertically, so that they remain in view.

You enable automatic resizing or repositioning in the Coordinates & Sizing theme of the Property List



Repositioning and resizing options

There are two lines, **Horizontal Sizing** and **Vertical Sizing**, for which you can assign three properties (None, Grow, and Move).

Here is the effect of these options:

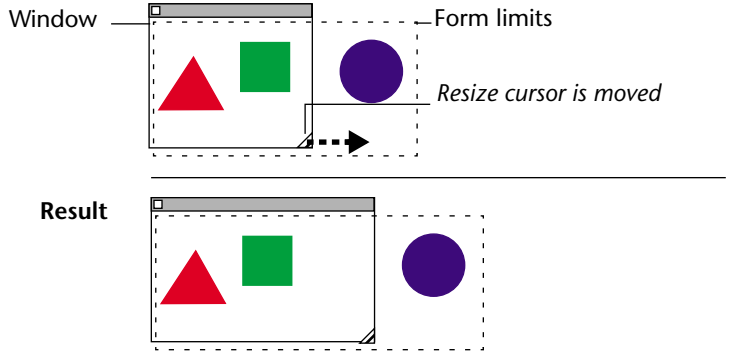
Option	Result
Horizontal Sizing: Grow	When the user resizes the width of the window, 4D applies the same percentage to the object's width.
Horizontal Sizing: Move	When the user resizes the width of the window, 4D moves the object left or right the same amount as the width increase.
Vertical Sizing: Grow	When the user resizes the height of the window, 4D applies the same percentage to the object's height.
Vertical Sizing: Move	When the user resizes the height of the window, 4D moves the object up or down the same amount as the height change.

The repositioning options enable the object to move in the specified direction to try to remain visible.

When the **None** option is used, the object remains stationary when the form is resized.

Side Pushers

In 4D, the right side and bottom of windows have become "pusher" splitters by default. This means that objects found to the right or below the limits of a window on screen are automatically pushed to the right or toward the bottom if the window is enlarged:



Note This does not work with windows that have scroll bars.

Using the Rulers

The Form editor rulers extend along the height and width of the form.

You can hide the rulers to increase your working area in the Form editor window. You can display the rulers again whenever necessary. To hide or display rulers, choose **Rulers** from the **Display** submenu in the **Form** menu or the Form editor contextual menu.

The rulers contain markers that show the position of the pointer when creating or resizing an object. While you are moving the object, the markers change to show the top, bottom, left and right sides of the object.

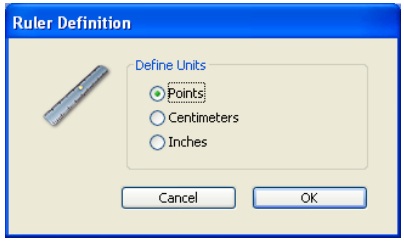
The object markers allow you to align other objects to the same position on the rulers.

You can change the units the rulers use to suit your preference.

► To define ruler units:

1 **Choose Ruler Definition from the Form menu.**

4D displays the Ruler Definition dialog box, shown below:



2 **Click the measurement unit you want to use.**

- Click **Points** to display rulers that provide measurement in points. One point is equal to the width of one pixel. There are 72 points in an inch.
- Click **Centimeters** to display metric scale rulers.
- Click **Inches** to display rulers that use feet and inches.

3 **Click OK.**

4D changes the measurement units to the scale you have selected. The objects' coordinates will also use the same units. For more information about this, see the section "Resizing Objects" on page 414.

Grouping Objects

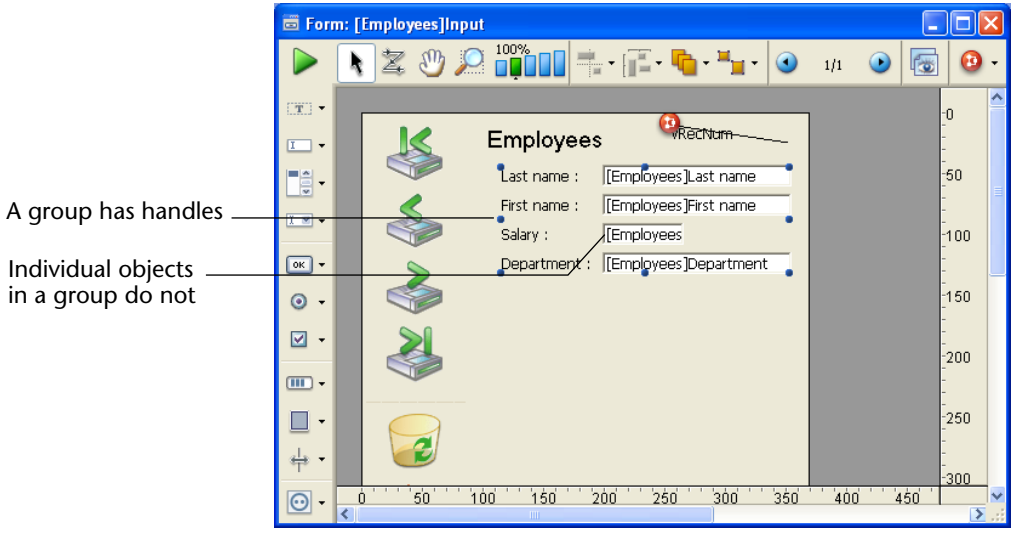
4D lets you group objects so that you can select, move, and modify the group as a single object.

Objects that are grouped retain their position in relation to each other. You would typically group a field and its label, an invisible button and its icon, and so forth. Groups can be part of other groups.

When you resize a group, all the objects in the group are resized proportionally (except text areas, which are resized in steps according to their font sizes).

Note Grouping is also used to synchronize scrollable areas. For more information, refer to the *4D Language Reference* manual.

The following illustration shows grouped objects:



You can ungroup a group of objects to treat them as individual objects again.

An active object that has been grouped must be ungrouped before you can access its properties or method.
However, it is possible to select an object belonging to a group without degrouping the set: to do this, **Ctrl+click** (Windows) or **Command+click** (Mac OS) on the object (the group must be selected beforehand).

Grouping only affects objects in the Form editor. When the form is executed, all grouped objects except for scrollable areas act as if they were ungrouped.

Note It is not possible to group objects belonging to different views and only those objects belonging to the current view can be grouped (see the "Using Object Views" paragraph on page 439).

► To group objects:

1 **Select the objects that you want to group.**

2 **Choose Group from the Object menu.**
OR
Click the Group button  **in the toolbar of the Form editor.**
4D marks the boundary of the newly grouped objects with handles. No handles mark the boundary of any of the individual objects within the group.

Now, when you modify the grouped object, you change all the objects that make up the group.

► To ungroup an object:

1 **Select the grouped object that you want to ungroup.**

2 **Choose Ungroup from the Object menu.**
OR
Click the Ungroup button (variant of the Group button) in the toolbar of the Form editor.
If **Ungroup** is dimmed, this means that the selected object is already separated into its simplest form.

4D marks the boundaries of the individual objects with handles.

Aligning Objects

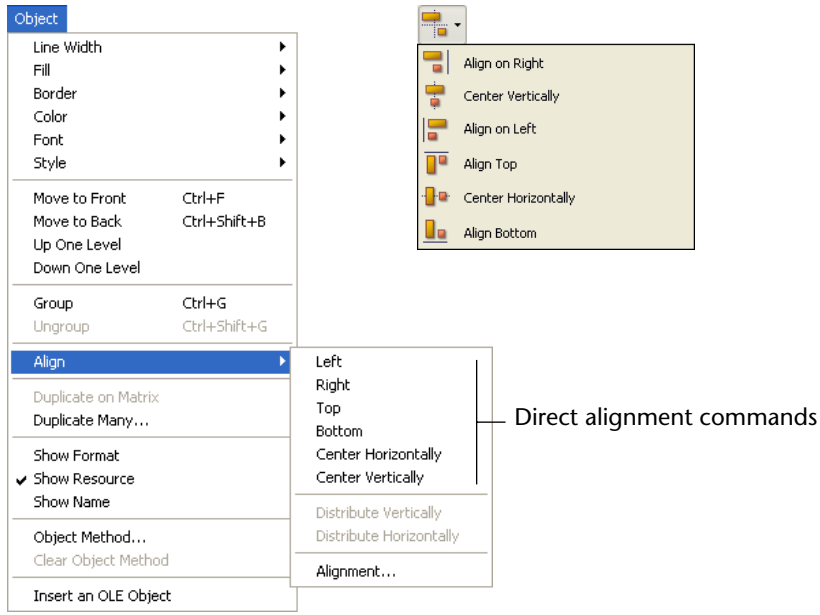
The Form editor's alignment tools and grid tools let you align objects to each other or using an invisible grid on the form.

■ When you align one object to another, you can align it to the top, bottom, side, or horizontal or vertical center of the other object. You can directly align a selection of objects using the alignment tools or apply more advanced alignment settings using the Alignment Assistant. The latter option allows you, for example, to set the object that will be used as the position reference and to preview the alignment in the form before applying it.

- When you use the invisible grid, each object can be aligned manually with others based on “noticeable” positions which are depicted with dotted lines that appear when the object being moved approaches other objects.

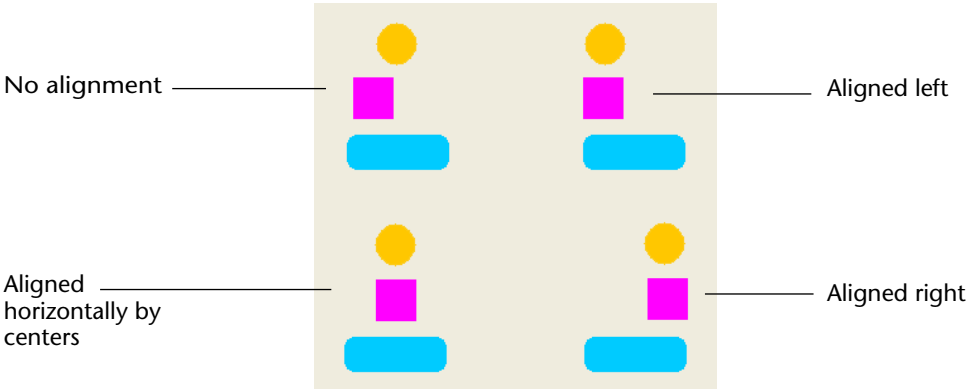
Using the Instantaneous Alignment Tools

The alignment tools in the toolbar and in the **Align** submenu of the **Object** menu allow you to quickly align selected objects.



When 4D aligns objects, it leaves one selected object in place and aligns the remaining objects to that one. This object is the “anchor.” It uses the object that is the furthest in the alignment’s direction as the anchor and aligns the other objects to that object. For instance, if you want to perform a right alignment on a set of objects, the rightmost object will be used as the anchor.

The figure below shows aligned objects:



► To align a set of objects:

1 **Select the objects you want to align.**
For more information on how to select objects, refer to "Selecting Objects" on page 410.

2 **Select the type of alignment that you want to use in the toolbar.**
OR
Choose an alignment menu command from the Align submenu in the Object menu.
OR
Choose an alignment menu command from the Align submenu in the contextual menu that is displayed when you right-click one object in the selection.
4D aligns the selected objects according to the alignment you have selected.

Using the Alignment Assistant

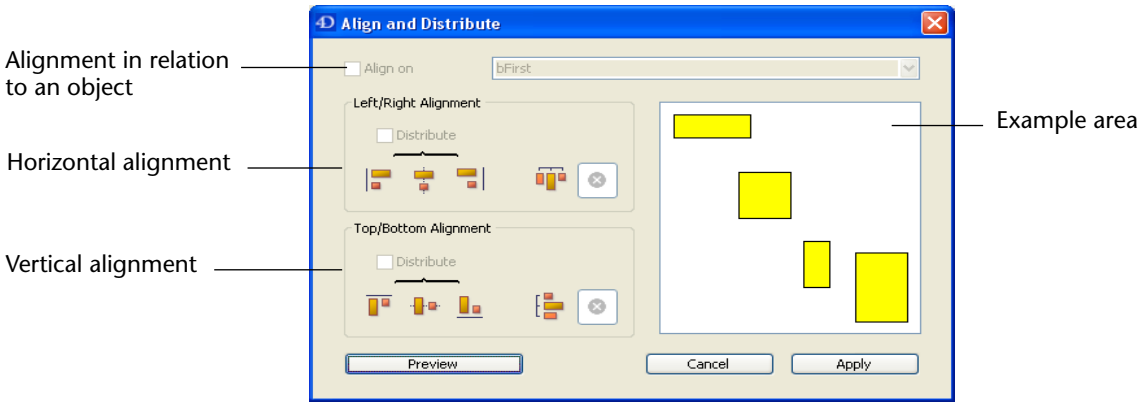
The Alignment Assistant allows you to perform any type of alignment and/or distribution of objects.

► To align objects using the Alignment Assistant:

1 **Select the objects you want to align.**
For more information on how to select objects, refer to "Selecting Objects" on page 410.

2 **Choose the Alignment command from the Align submenu in the Object menu.**
OR
Choose the Alignment command from the Align submenu in the Form editor contextual menu.

The following dialog box is displayed:



3 **In the "Left/Right Alignment" and/or "Top/Bottom Alignment" areas, click the icon that corresponds to the alignment you want to perform.**

The example area displays the results of your selection.

4 **To perform an alignment that uses the standard anchor scheme, click Preview or Apply.**

In this case 4D will perform an alignment using the standard anchor as described in "Using the Instantaneous Alignment Tools" on page 421.

OR

To align objects to a specific object, select the "Align on" option and select the object to which you want the other objects to be aligned from the object list.

In this case, the position of the reference object will not be altered.

You can preview the results of the alignment by clicking the Preview button. The objects are then aligned in the Form editor but since the dialog box does not go away, you can still cancel or apply the alignment.

Note This dialog box allows you to align and distribute objects in one operation. For more information on how to distribute objects, refer to "Distributing Objects" on page 425.

Using the Magnetic Grid

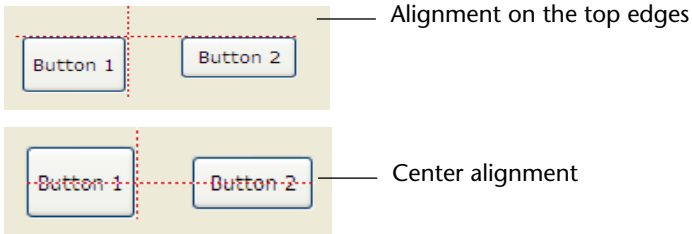
The Form editor provides a virtual magnetic grid that can help you place and align objects in a form.

Magnetic alignment of objects is based on their position in relation to each other. The magnetic grid can only be used when at least two objects are present in the form.

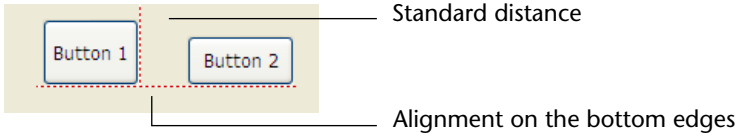
This works as follows: When you move an object in the form, 4D indicates possible locations for this object based on *noticeable alignments* with other form objects. A noticeable alignment is established each time that:

- Horizontally, the edges or centers of two objects coincide,
- Vertically, the edges of two objects coincide.

When this happens, 4D places the object at the location and displays a red line indicating the noticeable alignment taken into account:



Concerning the distribution of objects, 4D proposes a distance based on interface standards. Like with magnetic alignment, red lines indicate the noticeable differences once they are reached.



This operation applies to all types of form objects.

The magnetic grid can be enabled or disabled at any time using the **Turn Magnetic Grid On** command in the **Form** menu or in the editor contextual menu. It is also possible to set the activation of this feature by default on the **Form Editor** page ("Design Mode" theme) of the application Preferences (see the "Form Editor Page" paragraph on page 181).

You can manually activate or deactivate the magnetic grid when an object is selected by pressing the **Ctrl** (Windows) or **Control** (Mac OS) key.

Note The magnetic grid also influences the manual resizing of objects.

Distributing Objects

You can distribute objects so that they are set out with an equal amount of space between them. To do this, you can distribute objects using either the Distribute tools in the Tools palette or the Alignment Assistant. The latter allows you to align and distribute objects in one operation.

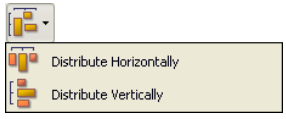
Note When the magnetic grid is on, a visual guide is also provided for distribution when an object is moved manually. For more information, refer to the "Using the Magnetic Grid" paragraph on page 423.

▶ To distribute objects with equal spacing:

1 **Select three or more objects and click the desired Distribute tool.**

For more information on how to select objects, refer to "Selecting Objects" on page 410.

2 **In the toolbar, click on the distribution tool that corresponds to the distribution you want to apply.**



OR
Select a distribution menu command from the Align submenu in the Object menu.
OR
Select a distribution menu command from the Align submenu in the contextual menu that is displayed when you right-click one object of the selection.

4D distributes the objects accordingly. Objects are distributed using the distance to their centers and the largest distance between two consecutive objects is used as a reference.

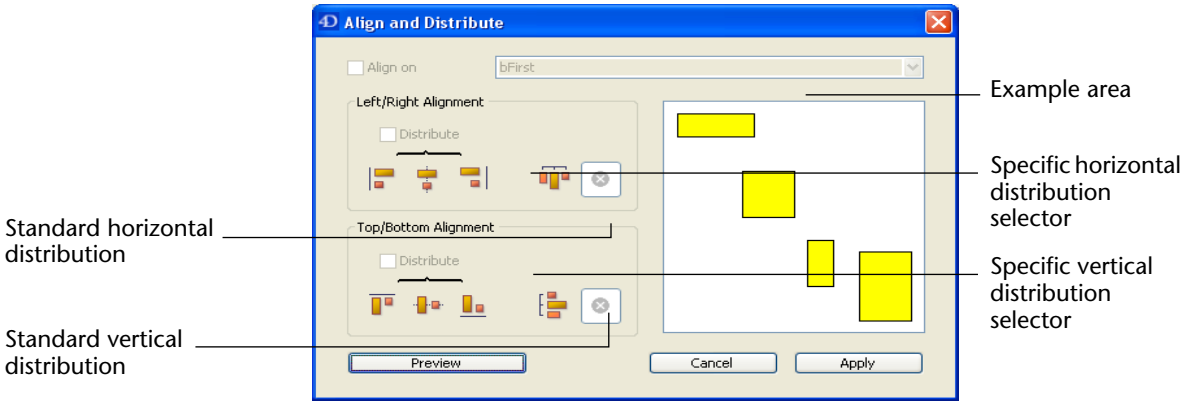
▶ To distribute objects using the Align and Distribute dialog box:

1 **Select the objects you want to distribute.**

For more information on how to select objects, refer to "Selecting Objects" on page 410.

2 **Choose the Alignment command from the Align submenu in the Object menu.**
OR
Choose the Alignment command from the Align submenu in the Form editor contextual menu.

The Alignment Assistant appears:



3 **In the Left/Right Alignment and/or Top/Bottom Alignment areas, click the standard distribution icon.**

The example area displays the results of your selection.

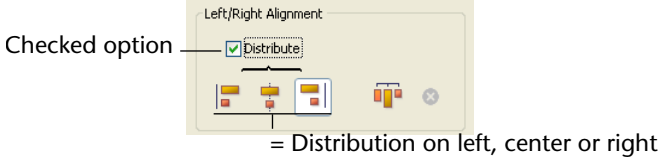
4 **To perform a distribution that uses the standard scheme, click Preview or Apply.**

In this case 4D will perform a standard distribution, as described earlier in this section.

OR

To execute a specific distribution, select the Distribute option (for example if you want to distribute the objects based on the distance to their right side).

This option acts like a switch. If the Distribute check box is selected, the icons located below it perform a different function.



Horizontally, the icons correspond to the following distributions: evenly with respect to **left sides**, **centers (hor.)** and **right sides** of the selected objects.

Vertically, the icons correspond to the following distributions: evenly with respect to **top edges**, **centers (vert.)** and **bottom edges** of the selected objects.

You can preview the actual result of your settings by clicking on the **Preview** button: the operation is carried out in the Form editor but the dialog box stays in the foreground. You can then **Cancel** or **Apply** the modifications.

Note This dialog box lets you combine object alignment and distribution. For more information about alignment, refer to the "Aligning Objects" paragraph on page 420.

Duplicating Objects

You can duplicate any object in the form, including active objects. Copies of active objects retain all the properties of the original, including name, type, standard action, display format, and object method.

You can duplicate an object directly using the Duplicate tool in the Tools palette or use the Duplicate Many dialog box to duplicate an object more than once. Also, using this dialog box, you can set the distance between two copies.

When duplicating a variable, you can use specific duplication features that allow you to include an automatic number in the copies' names. For more information on this point, refer to "Duplicating on a Matrix" on page 545.

► To duplicate an object:

1 **Select the object or objects that you want to duplicate.**

2 **Choose Duplicate from the Edit menu.**

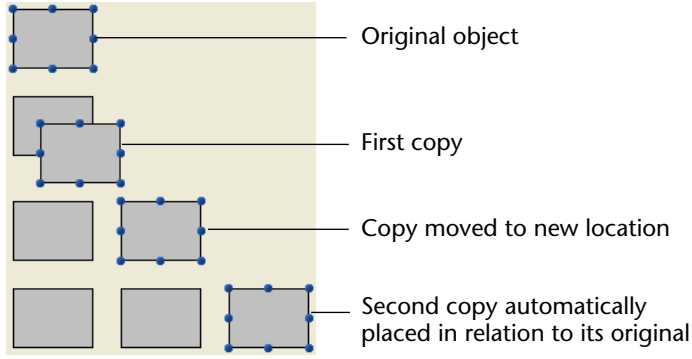
4D creates a copy of each selected object and places the copy in front and slightly to the side of the original.

3 **Move the copy (or copies) to the desired location.**

If you choose the **Duplicate** menu item again, 4D creates another copy of each object and moves it the exact same distance and direction from the first copy.

If you need to distribute copies of the object along a line, you should use the following procedure. Duplicate the original object, move the copy to another location in the form, and then duplicate the copy. The second copy is automatically placed in the same relation to the first copy as the first copy was in relation to the original object. Subsequent copies are also placed in the same relation to their originals.

The figure below shows how this relative placement of copies works:

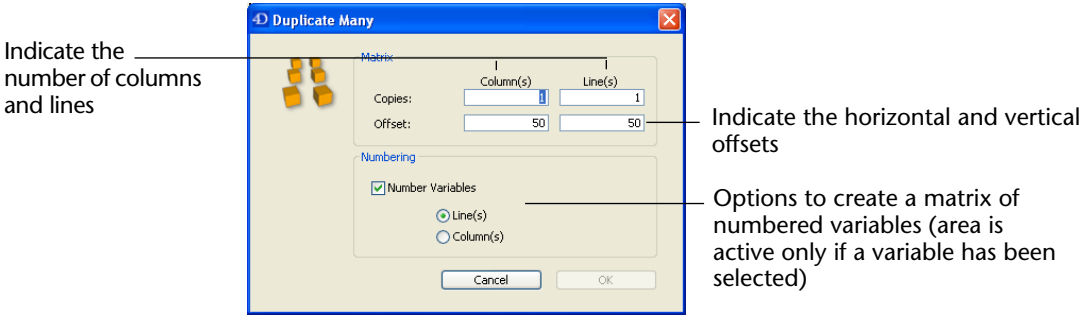


► To duplicate multiple objects using the Duplicate Many dialog box:

1 Select the object(s) to duplicate.

2 Choose Duplicate Many... from the Object menu.

The Duplicate Many dialog box appears:



3 In the upper area, enter the number of columns and lines (rows) of objects you want to get.

- For example, if you want three columns and two lines of objects, enter 3 in the Column(s) area and 2 in the Line(s) area.
- If you want three horizontal new copies of an object, enter 4 in the Column(s) area and leave the default value, 1, in the Line(s) area.

4 For lines and columns, define the offset that you wish to leave between each copy.

The value must be expressed in points. It will be applied to each copy, or copies, in relation to the original object.
For example, if you want to leave a vertical offset of 20 points between each object and the height of the source object is 50 points, enter 70 in the column's "Offset" area.

5 **If you wish to create a matrix of variables, select the Number Variables option and select the direction in which the variables are to be numbered, either by line(s) or by column(s) (otherwise go to step 6).**

This option is active only when the selected object is a variable. For more information on this option, refer to "Duplicating on a Matrix" on page 545.

6 **Click the OK button.**

The number of columns and lines of the specified object(s) are created.

Copying Form Objects

You can copy all, some, or one of the objects on a form using the **Copy** menu item. You can use the copied objects in another form in the database, in another database or in an object library (see the "Using Object Libraries" paragraph on page 461).

Like objects duplicated using the **Duplicate** command, each object copied using the **Copy** command is a complete copy of the original object and retains all the properties of the original. Copies of active objects such as fields and buttons retain all the properties of the original including name, type, action, display format, and the method associated with the object.

Objects copied using the **Copy** command are copied to the Clipboard while objects duplicated with the **Duplicate** command are reproduced on the current form page.

Note A method that is copied with an object and pasted in a different form may lose its meaning unless it is updated. This is the case, for example, if the method refers to a specific field and the new form does not have this field.

► To copy all the objects in a form and paste them into another location:

1 **Choose Select All from the Edit menu or from the contextual menu or use the standard shortcut (Ctrl+C / Command+C).**

4D selects every object on the current form page.

Note This command does not select objects which belong to views that are not displayed.

2 **Choose Copy from the Edit menu.**

4D places a copy of the form on the Clipboard.

3 **Display another page of the form, another form or an object library**.

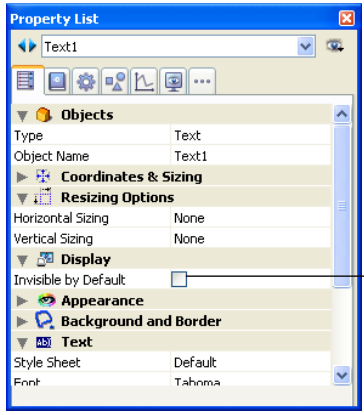
4 **Choose Paste from the Edit menu or from the contextual menu or use the standard shortcut (Ctrl+V / Command+V).**

The selection is pasted into the destination object.

Note The output control lines are not associated with objects, so they must be repositioned on the new form. For information about working with output control lines, see "Moving Output Control Lines" on page 645.

Making Objects Invisible by Default

You can associate the **Invisible by Default** property with most form objects:



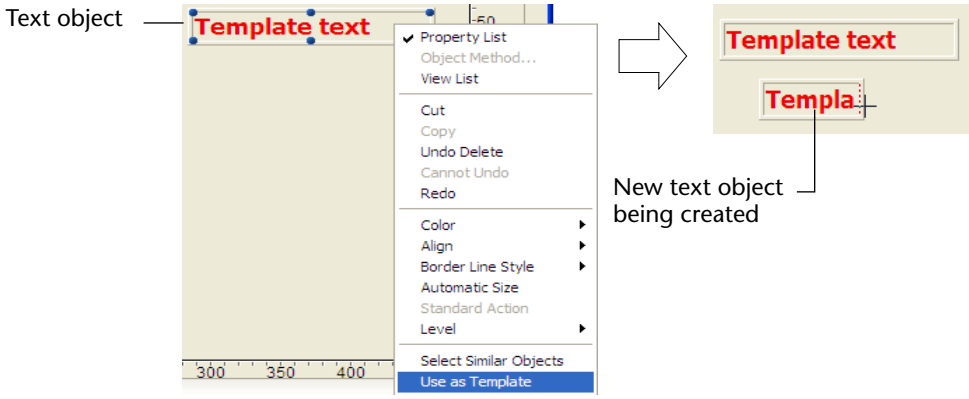
This property simplifies dynamic interface development. In this context, it is often necessary to hide objects programatically during the On load event of the form then to display certain objects afterwards. The **Invisible by Default** property allows inverting this logic by making certain objects invisible by default. The developer can then program their display using the SET VISIBLE command depending on the context. For more information about this command, refer to the *4D Language Reference* manual.

Note Do not confuse this property with the **Invisible** property in pop-up menus, which allows not drawing the object while still leaving it active.

Setting an Object Template

The **Use as Template** command, available in the contextual menu of the editor when an object is clicked, saves the clicked object as a template. This template and its properties will then be used when creating all objects of the same type.

In the following example, the command is applied to a Text object. All Text objects created afterwards will use the properties of the designated object by default:



A customized template can be used for each object type.
There is a single customized template per object type for the entire database.

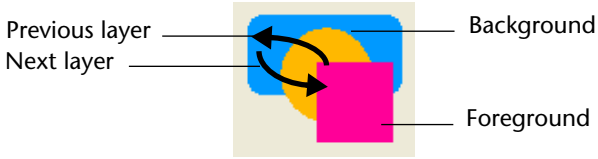
The template saves all object properties set when the command was executed, with the exception of fixed coordinates and the object method (if applicable).

To delete an object template, you must create a new one. You can also remove all object templates (and go back to the standard template) using the **Clear all custom templates** button on the **Form Editor** page ("Design Mode" theme) in the application Preferences. For more information, refer to the "Form Editor Page" paragraph on page 181.

Layering Objects

You will sometimes have to rearrange objects that are obstructing your view of other objects in the form. For example, you may have a graphic that you want to appear behind the fields in a form. 4D provides four menu items, **Move to Back**, **Move to Front**, **Up One Level** and **Down One Level** that let you "layer" objects on the form.

The figure below shows objects in front of and behind other objects.

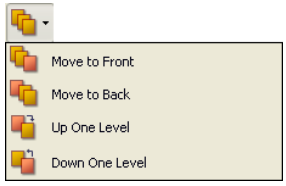


Note When several objects are superimposed, the **Ctrl+click / Command+click** shortcut can be used to select each object successively by going down a layer with each click.

► To move one or several objects to the back:

1 **Select the object(s) that you want to move to the back.**

2 **Select the Move to Back command in the toolbar.**



OR
Choose Move to Back from the Object menu.
OR
Choose Move to Back from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse.

4D moves the selected object or objects behind all the other form objects.

► To move an object one level to the back:

1 **Select the object or objects that you want to move to the back.**

2 **Select the Down One Level command in the toolbar.**
OR
Choose Down One Level from the Object menu.
OR
Choose Move Down One Level from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse.

4D moves the selected objects down one level.

► To move an object to the front:

1 Select the object or objects that you want to move to the front.

2 Select the Move to Front command in the toolbar.
OR
Choose Move to Front from the Object menu.
OR
Choose Move to Front from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse.

4D moves the selected object or objects in front of all the other objects.

► To move an object one level to the front:

1 Select the object or objects that you want to move to the front.

2 Select the Up One Level command in the toolbar.
OR
Choose Up One Level from the Object menu.
OR
Choose Move Up One Level from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse.

4D moves the selected objects up one level.

Deleting Objects

You can delete any object on a form. If you want, you can place a copy of the deleted object on the Clipboard. Objects placed on the Clipboard can later be pasted to new locations in the form. If the Property List window is open when you delete an object, the deletion will affect the contents of this window.

► To delete an object:

1 Select the object or objects you want to delete.

2 Choose Clear from the Edit menu.
OR
Press the Backspace key (Windows) or the Delete key (Mac OS) on your keyboard.

4D deletes the selected object or objects.

To copy an object to the Clipboard, choose **Cut** from the **Edit** menu. 4D removes the selected object or objects and places a copy on the Clipboard. The **Cut** command works even when the Property List window is open.

If you change your mind, choose **Undo** from the **Edit** menu or from the editor's contextual menu before performing another action. 4D restores the deleted object or objects.

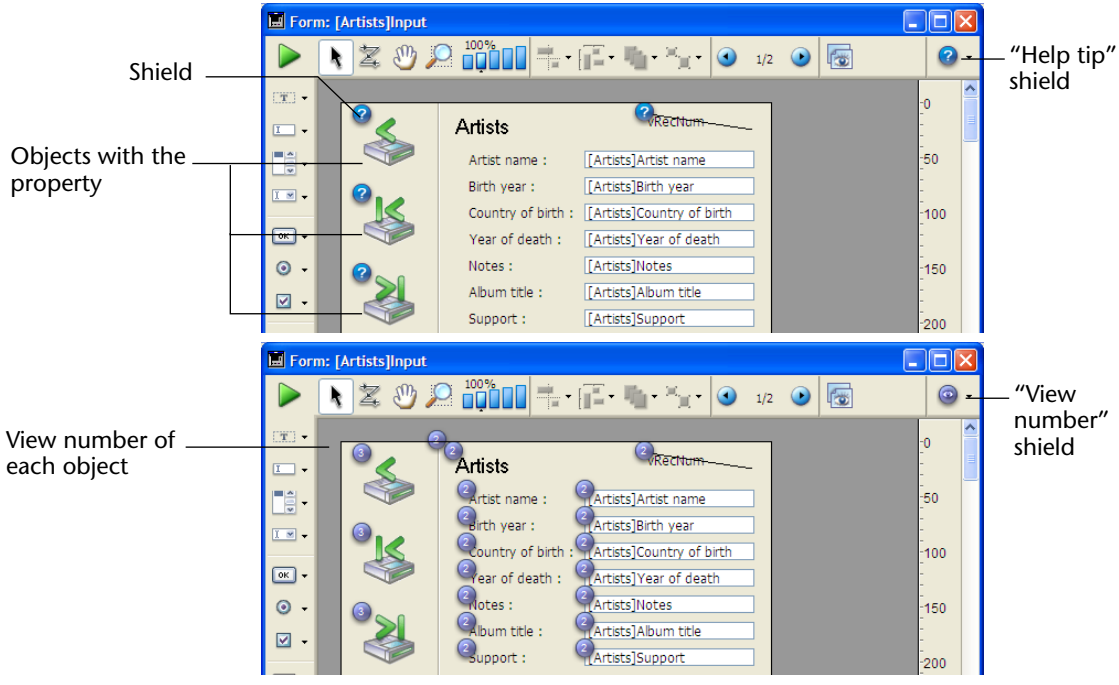
Using Shields

The Form editor allows using *shields* to make viewing object properties easier.

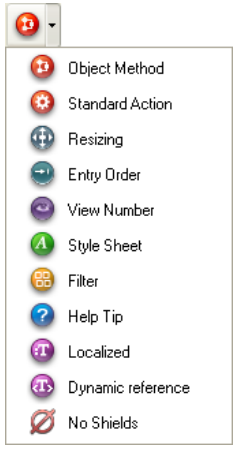
This function works as follows: Each shield is associated with a property (for example, **Tips**, which means "has an associated tip"). When you activate a shield, 4D displays a small icon (shield) in the upper left of each object of the form where the property is applied.

For "true/false" properties (such as Tips), the shield is only displayed if the object has the property.

For "value" type properties (such as "View number"), the shield displays the value of the property for each object.



To activate a shield, click the shield selection button until the desired shield is selected. You can also click on the right side of the button and select the type of shield to display directly in the associated menu:



If you do not want to display shields, select **No Shields** in the selection menu.

Note You can set which shields to display by default on the **Form Editor** page ("Design Mode" theme) of the application Preferences. For more information, please refer to the "Form Editor Page" paragraph on page 181.

The following is a description of each type of shield:

Icon	Name	Description
	Object Method	Displayed for objects with an associated object method
	Standard Action	Displayed for objects with an associated standard action
	Resizing	Displayed for objects with at least one resizing property, indicates the combination of current properties

 Expand horizontally

 Move horizontally

 Expand vertically

 Move vertically

 Expand horizontally, Expand vertically

 Expand horizontally, Move vertically

Icon	Name	Description
	 Move horizontally, Expand vertically	
	 Move horizontally, Move vertically	
	Entry Order	Displayed for enterable objects, indicates the number of entry order
	View Number	Displayed for all objects, indicates the view number
	Style Sheet	Displayed for objects with an associated style sheet
	Filter	Displayed for enterable objects with an associated entry filter
	Help Tip	Displayed for objects with an associated tip
	Localized	Displayed for objects whose label comes from a reference (label beginning with ":"). The reference can be of the resource (STR#) or XLIFF type.
	Dynamic reference	Displayed for objects containing a dynamic reference to a field, table or variable (syntax of the "<label>" type)
	No shields	No shields appear

Optimizing the Appearance of Text and Picture Objects

You can resize static text areas and pictures for optimal display based on their current graphic characteristics (font size, style, etc.).

To do this:

1 **Right-click on the object and choose the Automatic Size command from the contextual menu.**
OR
Hold down the Ctrl key (under Windows) or Command key (under Mac OS) and click on the lower right corner of the object.

- Static text areas, check boxes, radio buttons and buttons will be resized so that their contents correspond exactly to their boundaries.
- Fields and variables will be resized so that their height is at least equivalent to that of the current font and their width is 100 points — if the size of the object is greater than these minimum sizes, the command will have no effect.
- Pictures or picture buttons will be displayed using the original size of the source picture.

- Scrollable areas and hierarchical lists will be displayed with a height in keeping with that of the current font.
- Combo boxes, pop-up/drop-down lists and hierarchical pop-up menus will be displayed with their default minimum height if the current height is insufficient.

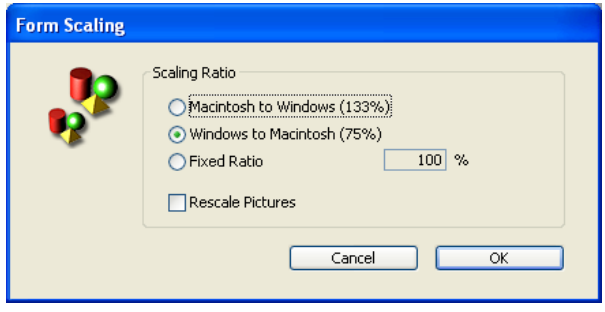
Scaling a Form

The Form editor includes a feature for rescaling form objects so that they look good when a database is transported to another platform.

Form objects created under Mac OS will look smaller when viewed under Windows, and vice versa — even though the objects are actually the same size. This is because the Windows screen resolution is about 25% greater than the Macintosh resolution. For instance, 12-point text on a Macintosh will appear as 9-point text under Windows. If the font size is just large enough under Mac OS, it may be too small under Windows. Conversely, if a font size under Windows is adequate, it may be too large under Mac OS.

To compensate for screen resolution differences, you need to rescale objects. With the **Form Scaling** item in the **Form** menu you can proportionally resize all the form objects in one operation.

When you choose **Form Scaling**, the Form Scaling dialog box appears.



You can choose among the following options:

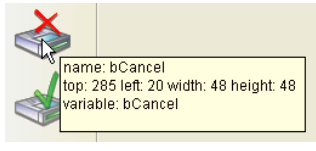
- **Macintosh to Windows (133%)**: This option is the default option when you use 4D under Mac OS. Use this option when you want to resize a form that was created according to the Macintosh screen resolution so it will look like it was created according to the Windows screen resolution. To do so, the program increases the size of all the form objects by one-third. For instance, 9-point text will become 12-point text.

- **Windows to Macintosh (75%)**: This option is the default option when you use 4D under Windows. Use this option when you want to resize a form that was created according to the Windows screen resolution so it will look like it was created according to the Macintosh screen resolution. To do so, the program decreases the size of all the form objects by one-quarter. For instance, 12-point text will become 9-point text.
- **Fixed Ratio**: This option lets you resize a form using the percentage you type in the "%" enterable area. With this option you can resize a form so it will look good on any unusual screen resolution you may encounter on either the Macintosh or Windows platform. You can also use this option to change the size of all the form's objects for the platform you are using. For example, it you want to double the size of all objects, enter 200%; if you want to halve the size, enter 50%.
- **Rescale Pictures**: This option is not selected by default. Usually, decreasing or increasing the size of bitmapped pictures does not provide good results from a cosmetic point of view. For this reason, the program does not resize any static pictures in a form unless you select this option. Instead it moves them to their new "center relative" positions. If you know that rescaling bitmaps will produce pleasing results or if you use non-bitmapped pictures, you may chose to rescale the pictures.

When you have selected your options, click **OK** to resize the form, otherwise click **Cancel**. If you resize a form by mistake or with the wrong option, choose **Undo** from the **Edit** menu to recover the form as it was before the rescaling.

Information about Objects on Forms being Executed

When a form is being executed, a shortcut can be used to obtain various useful information about the objects it contains (name, coordinates, etc.). The information is displayed as a help tip that appears when you hold down **Ctrl+Shift** (Windows) or **Command+Shift** and move the cursor over an object:



This information is available for each object displayed in a form when the Design environment is open.

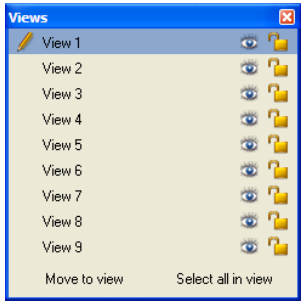
Using Object Views

You can use views in 4D forms. This new function makes building complex forms easier by allowing the distributing of objects within different views that can then be hidden, displayed or locked. Distributing objects can be carried out, for example, according to their type (fields, variables, static objects, etc.). All object types, including subforms and plug-in areas, are included in views.

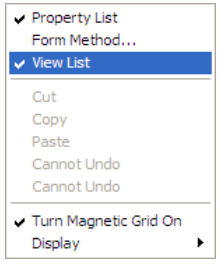
How it Works

There are 9 views available per form, named View 1 to View 9 by default (these names can be changed). Each view can be displayed or hidden and locked.

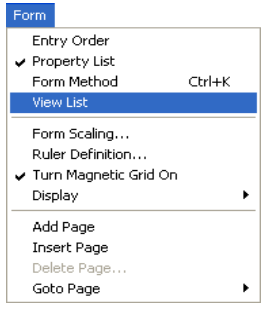
View management is done through the views palette:



To display this palette, click the view button in the window toolbar  or choose the **View List** command in the editor contextual menu or in the **Form** menu:



Contextual menu



Here are a few rules for working with views:

- **Context of use**: Views are a purely graphic tool that can only be used in the Form editor; you cannot access views programmatically or in the Application environment.

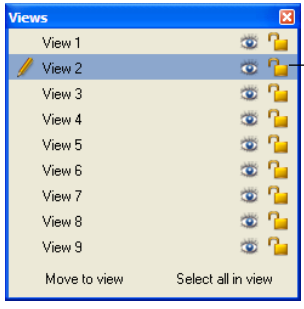
- **Views and pages**: Objects of the same view can belong to different form pages; only objects of the current page (and of page 0 if it is visible) can be displayed, regardless of the view configuration.
- **Views and levels**: Views are independent of object levels; there is no display hierarchy among different views.
- **Views and groups**: Only objects belonging to the current view can be grouped.

Note In forms created by default or using the Form Wizard, the title is placed in View 2 and the action buttons (picture buttons) in View 3.

Placing an Object in a View

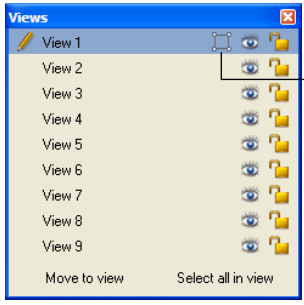
An object can only belong to a single view. Any object created in a form is placed in the current view. By default, View 1 is selected; consequently, all objects are placed in the first view of the form.

To create an object in another view, simply select the view beforehand in the palette by clicking its line:



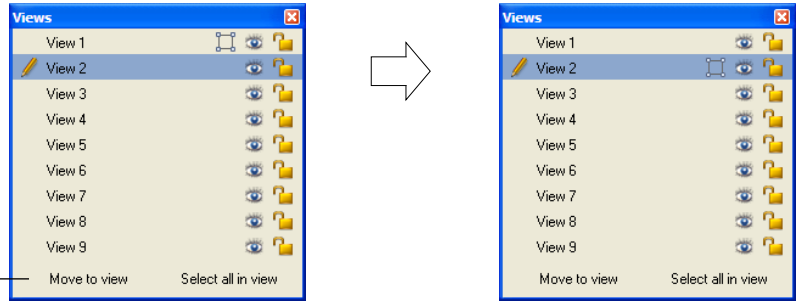
It is also possible to move one or more object(s) from one view to another. To do this, select the object(s) in the form whose view you wish to change.

The view list indicates, using a symbol, the view to which the selection belongs:



Note The selection can contain several objects belonging to different views.

Then, simply select the destination view of the selection and click **Move to view**. The selection is then placed in the new view:

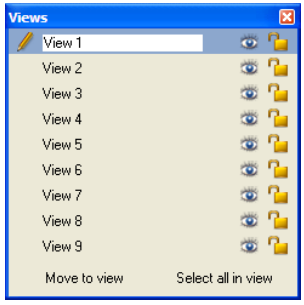


Note You can display the view in which each object is found using the 4D shields. For more information on this, please refer to the "Using Shields" paragraph on page 434.

Renaming Views

You can rename each of the 9 views, for example, if you want to give a name that describes the object contents. You can rename the 9 views differently in each database form.

To rename a view, you can use either **Ctrl+click** (Windows) or **Command+click** (Mac OS) on the view name, or double-click the view name (the selected view in this case). The name then becomes editable:



Working with Views

Once you put each object in a view, you can use the views palette to:

- Select all objects of the same view in just one click,
- Display or hide objects for each view,
- Lock the objects of a view.

Select All Objects of a View

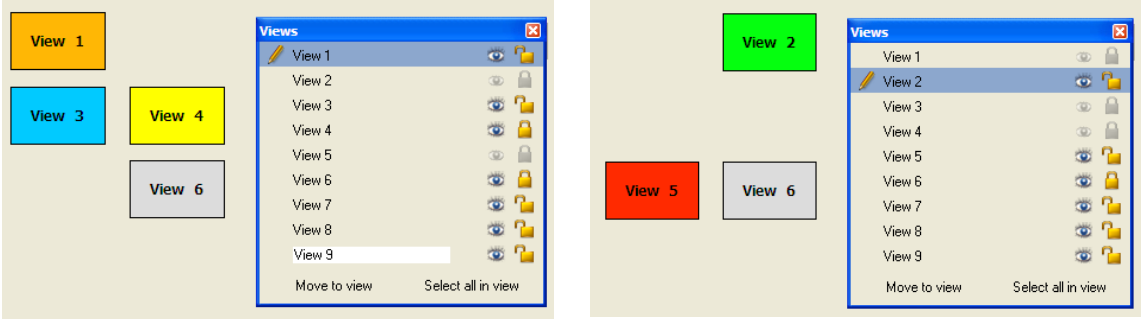
You can select all objects belong to the same view in the current page of the form. This function is useful for applying global changes to a set of objects.

To do this, select the view in which you wish to select all the objects and click **Select all in view**.

Display or Hide Objects of a View

You can display or hide objects of a view at any time in the current page of the form. This function allows, for example, concentrating on certain objects when editing the form.

By default, all views are displayed, as indicated by the display icon next to each view in the views palette. To hide a view, click this icon. It is then dimmed and objects of the corresponding view are no longer displayed in the form:



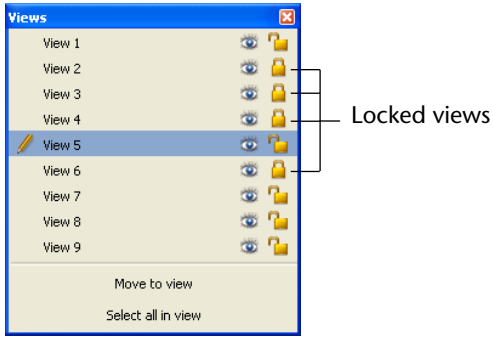
Note The current view (selected in the list of views) cannot be hidden.

To display a hidden view, simply select it or click on the display icon again.

Locking Objects of a View

You can lock objects of a view, which prevents them from being selected, changed or deleted from the form. Once locked, an object cannot be selected by a click, a rectangle or the **Select Similar Objects** command of the contextual menu. This function is useful for preventing manipulation errors.

By default, all views are unlocked, as indicated by the  icon next to each view in the views palette. To lock the objects of a view, click the above icon. The padlock is shut, which means that the view is locked:



Note The current view (selected in the list of views) cannot be locked.

To unlock a view, simply select it or click on the lock icon again.

Changing the Appearance of Objects

You can change the appearance of any object in a form. Each object has its own platform interface and appearance setting.

For any object that uses text (a field, a text area, a button, and so forth), you can change the following attributes:

- Platform interface
- Appearance
- Font
- Style

- Size
- Alignment within the object's area.

For any object that uses lines, fill patterns, or colors, you can change the following attributes:

- Line width
- Fill pattern
- Border pattern
- Foreground and background color.

Platform

You can set the platform interface on an object-by-object basis. To do this, you have the following choices:

- **Inherited from Form**: The platform interface for the object is the same as the platform interface of the form. The platform interface of the form is set in the form properties.
- **System**: Regardless of the platform set at the form level, the object is drawn according to the platform where the application is running.
- **Printing**: Regardless of the platform set at the form level, the object is drawn so as to be suitable for printing.

Note If the database was converted from an earlier version of 4D, additional platform interface properties are available. For more information about platform interface properties, refer to "Platform" on page 396.

► To set the platform interface of an object:

1 **Display the object properties in the Property List.**

2 **Expand the "Appearance" theme.**

3 **Select the platform interface from the drop-down list.**

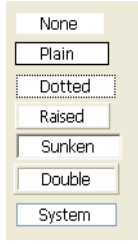
Border Line Style

You can set the border line style for most objects. To do this, you have the following choices available:

- **Transparent**: Objects appear with no border.
- **Plain**: Objects appear framed with a continuous 1-pt. border line.
- **Dotted**: Objects appear framed with a dotted 1-pt. border line.
- **Raised**: Objects appear framed with a 3D effect (raised).

- **Sunken**: Objects appear framed with a sunken 3D effect.
- **Double**: Objects appear framed with a double line, i.e., two continuous 1-pt. lines separated by a pixel.
- **System**: Under Windows, objects appear with a blue 1-pt. line. Under Mac OS, the equivalent of the Sunken option is used.

  The following example compares these styles under Windows with the "System" platform interface:



Note The result is different when the "Printing" platform interface option is used.

For more information about the effects of these options on various form objects, refer to "Buttons" on page 551 and "Check Boxes" on page 572.

► To set a border line style:

1 **Display the object properties in the Property List.**

2 **Select the border line style from the drop-down list in the "Background and Border" theme.**
OR
Choose an option from the Border Line Style submenu of the contextual menu (right-click on the object).

Working with Static Text Areas

Static text areas include both the actual text areas and the group boxes (text areas associated with a graphic border).

In addition to the standard properties, you can modify the following specific aspects in text areas:

- Add to or edit the text,
- Set the font attributes for the area.

Creating and Editing Text Areas

Use text areas to provide labels, titles, and descriptions in your form.

Text you create in a static text area is different from the text contained in a Text field. A Text field contains data stored in the database. The contents can be different in each record. A static text area is a graphic object; it is not active. Text in a text area remains the same whenever the form is displayed.

There are three exceptions to this rule:

- You can embed field names or variable names in static text areas. When the text area is displayed or printed, the values of the fields or variables from the current record are substituted. Use embedded fields and variables to create mail-merge documents and in report headers and footers. For complete details, see the section "Creating Mail-Merge Documents" on page 661.

- You can integrate dynamic table or field references in text areas. When you place dynamic labels in your forms, they are automatically updated throughout the database when you modify a table's or a field's name using the Structure editor or the SET TABLE TITLES and SET FIELD TITLES 4D commands.

  To insert a dynamic table name: <**?[TableName]**> or <**?[2]**> (the table's creation order number, meaning the second table created).

  To insert a dynamic field name: <**?[TableName]FieldName**> or <**?[2]3**> (the table's and field's creation order number), or even <**?3**> (the field's creation order number) for the current table's field.

  Please note that table and field numbers correspond to their creation order. You can therefore add or rename tables and fields without modifying the dynamic reference system. The actual contents of the text area can be displayed using the **Show Name** and **Show Format** menu commands from the **Object** menu.

Note You can automatically insert dynamic references using the Options page of the Form Wizard. For more information about this point, refer to "Display Options" on page 358.

- You can integrate text coming from an XLIFF translation file or an STR# resource into text areas. For more information about this point, please refer to appendix C "XLIFF Architecture"on page 1293. When you choose **Show Resource** in the **Object** menu, the referenced text is displayed according to the context.

The Form Wizard automatically creates text areas that contain field labels for the fields and, optionally, a title for the form. You can modify these labels as you would any other text area that you add to the form.

As you draw a text area, it snaps to a size that reflects the font size.

After you create a text area, the text box displays a sample text that is selected:

Type text in the text area. When the text you are typing reaches the edge of the text area, 4D automatically wraps the text to the next line in the area.

Note You can automatically adapt the size of a text area to its contents. Refer to "Optimizing the Appearance of Text and Picture Objects" on page 436.

If you enter more text than the area can display, the text is not visible until you resize the text area.

► To modify text in a text area or group box:

1 **Click on the tool icon for creating text areas / group boxes.**

2 **Select the text you want to modify or click in the area.**

OR

1 **Select the area and press Enter.**

The area switches directly to editing mode.

Use the standard text editing operations to edit the text.

Setting Text Attributes

You can set text attributes for text objects in the object properties of the Property List or the **Font** and **Style** submenus of the **Object** menu.

► To set text attributes using the **Font** and **Style** submenus:

1 **Select the object(s) whose text attributes you want to change.**

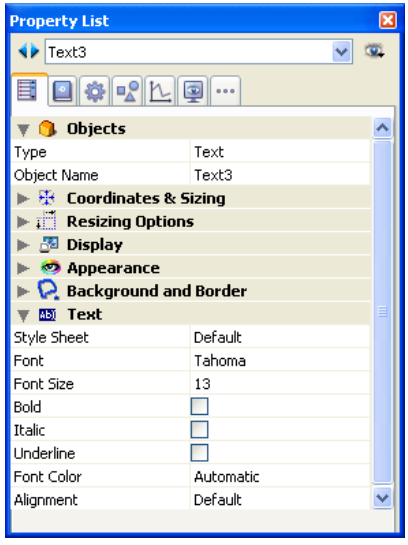
2 **Make the appropriate selections from the Font and Style submenus of the Object menu.**

► To set text attributes using the Property List:

1 **Display the properties in the Property List.**

2 **Expand the "Text" theme.**

The current text properties are displayed:



3 **Make the appropriate font, font size, and font style selections.**
OR
Choose a style sheet from the Style Sheet drop-down list.

For more information about style sheets, please refer to the chapter "Style Sheets" on page 923.

4 **Choose a font color if necessary.**

The font color is equivalent to the foreground color (see the "Foreground and Background Colors" paragraph on page 451.

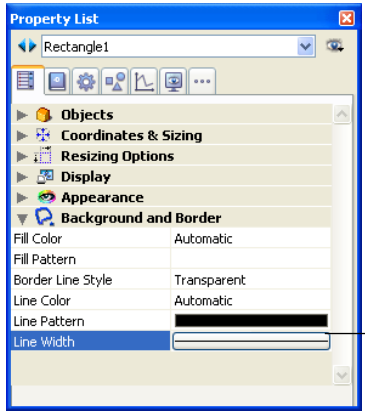
5 **Choose an alignment.**

The alignment determines the location of the text in the area that contains it.

Line Widths

4D lets you specify different widths for lines and objects that have lines such as ovals, grids, and rectangles.

You can specify line widths using either the **Line Width** menu item of the **Object** menu or in the Property List.



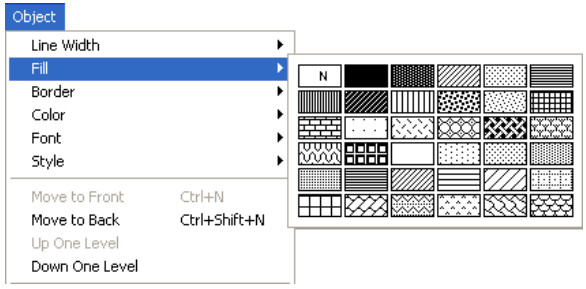
Choose one of the line widths. The first choice is the hairline, i.e., the thinnest line that can be printed by the printer, followed by 1, 2, 3 or 4 point lines.

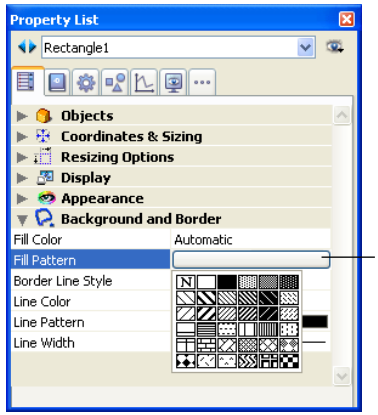
If you choose **Other**, 4D displays a dialog box in which you can specify any line width up to 20 points.

Fill Patterns

You can apply a fill pattern to any two-dimensional graphic object in the form such as an oval, a rectangle, a line, a grid object, and the enclosed area of a text object.

You can choose the fill pattern using either the **Fill** menu item in the **Object** menu or in the Property List:

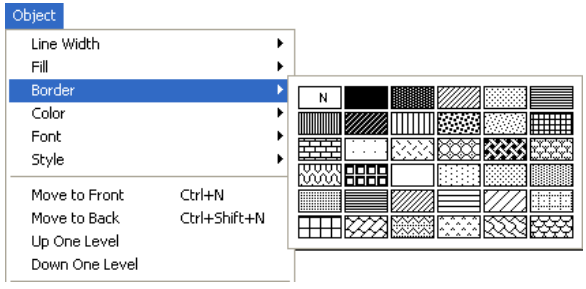


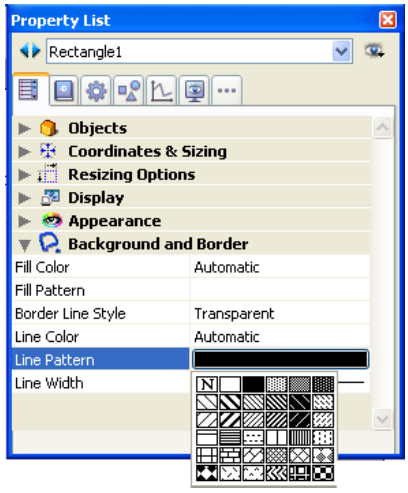


Border Patterns

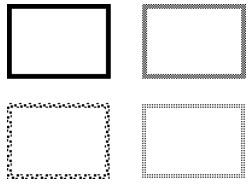
You can set patterns for the borders of any object in the form that has a border — such as an oval, a rectangle, or a grid object. The border patterns available are the same as the fill patterns. The appearance of the border also depends on the line width you have specified for the border.

You can set the border pattern using either the **Border** item of the **Object** menu or the Property List.





The following illustration shows examples of borders with patterns.



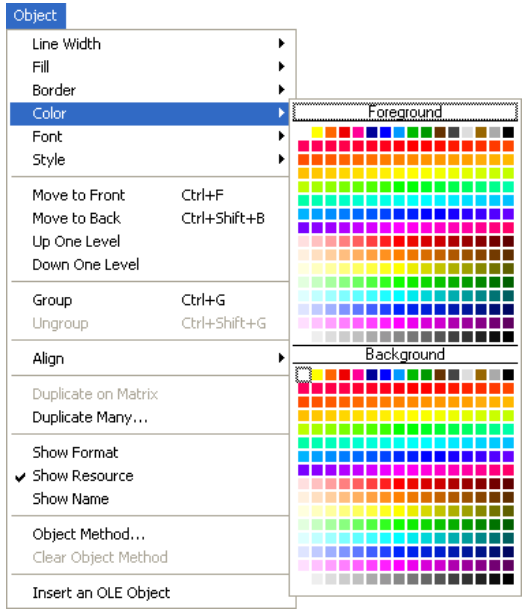
Foreground and Background Colors

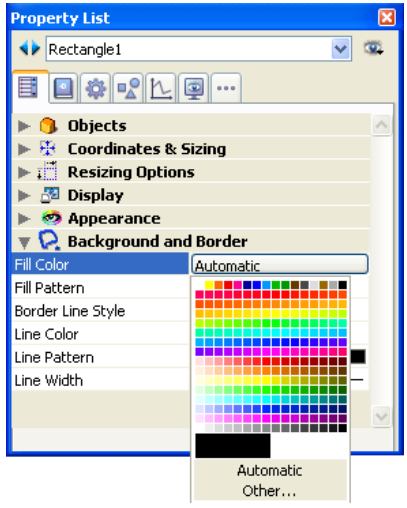
4D lets you add colors to objects for display on a color monitor or (if your printer supports color) for color printing. By combining colors and patterns your possibilities are virtually endless.

You can specify different colors for foreground pixels and background pixels. On a color display, the mixing of foreground and background colors allows you to create custom tints and shades.

In the Property List, the background color is called Fill Color and the foreground color is called Line Color. In the case of text objects, the Font Color corresponds to the foreground color.

You can set foreground and background colors using either the **Color** item in the **Object** or contextual menu, or using the Property List.





• If you select the **Automatic** option for foreground and background colors, the colors used will be the colors set in your OS.
You can set the **Automatic** option by selecting Foreground or Background in the **Color** submenu of the **Object** or contextual menu.

• The Property List allows you to use the system color chooser to set a color by selecting **Other**.

Inserting a Static Picture in a Form

You can use two types of static pictures in your forms: **independent pictures** stored directly with the forms or **picture references** stored outside the forms.

For better optimization, using pictures inserted by reference is generally recommended when pictures are used in several different places since it means that the picture will only be stored once. In addition, there is a link between the picture and each of its references so if you modify the source picture, all of its occurrences are automatically modified as well, wherever it is used in the database.

In 4D, pictures inserted by reference have the “Library Picture” type and independent pictures have the “Static Picture” type (displayed in the Type field of the Property List). To insert pictures by reference, you must either use the picture library (see the “Inserting via the Picture Library” paragraph on page 454), or drag and drop a picture stored in the Resources folder of the database (see the “Automatic Referencing of Picture Files” paragraph on page 456).

If you place a picture on page 0 of a multi-page form, it will appear automatically as a background element on all pages. You can also include it in an inherited form, applied in the background of other different forms. This way, your database will run faster than if the picture was pasted into each page.

You can assign properties to static pictures that are inserted into forms. For example the Replicated display property can be used to set a small picture as the background picture of the form: the picture is replicated as many times as necessary to fill in the page of the form.

Inserting Pictures

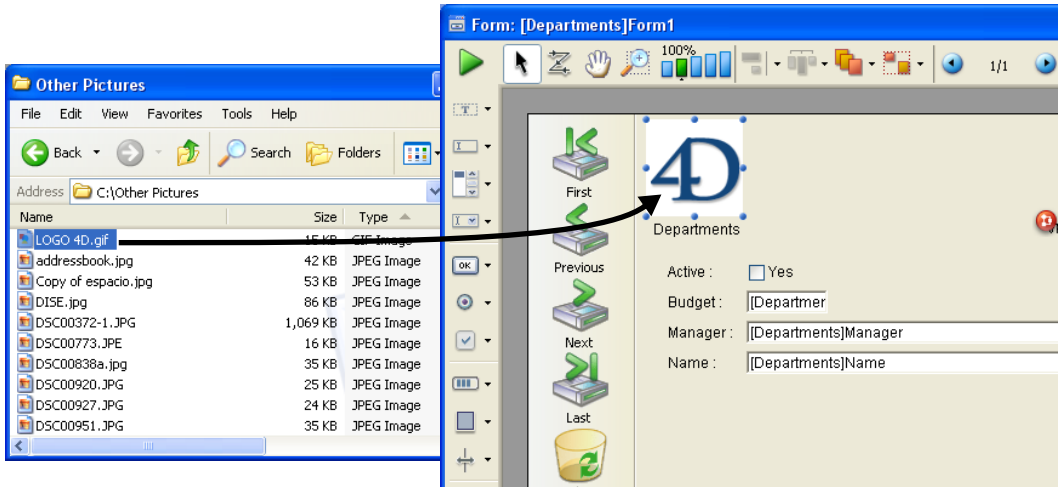
There are three ways to insert static pictures into your forms: by pasting a picture from the clipboard, by dragging and dropping a picture file or by dragging and dropping a picture from the picture library.

Inserting by Copy-paste or Drag and Drop

► To paste or drop a static picture into a form:

1 Copy a picture into the clipboard and paste it into the form.
OR

Select a picture in an external application (picture file in a system window, Web browser, other application, etc.) and insert it into the form using drag and drop.



The original picture must be stored in a format managed natively by 4D (4D recognizes the main picture formats: JPEG, PNG, BMP, SVG, GIF, etc.). It is automatically pasted into the target area in its native format.

Notes

- This also works with library pictures, pictures associated with objects (for example a 3D picture button) and fields or variables in Application mode.
- You can insert a picture reference by dragging and dropping a file stored in the Resources folder of the database (see the "Automatic Referencing of Picture Files" paragraph on page 456).

2 Position the picture and set its properties as needed.

Inserting via the Picture Library

For more information about how the picture library works, please refer to the chapter "Picture Library" on page 891.

► To place a picture from the picture library on a form:

1 Open the Picture Library in the Tool Box and click on the name of the desired picture.

2 Drag it from the Picture Library to the form.

- If the picture you drag is defined as a table of thumbnails, it will automatically be inserted as a picture button (or picture pop-up menu if you hold down the **Shift** key when dragging the picture). If you want to insert it as a static or library picture, press the **Alt** key (Windows) or the **Option** key (Mac OS) when dragging the picture.
- By default, the inserted picture will be of the **Library Picture** type, i.e., 4D will maintain the link with the original picture of the library (insertion by reference). If you want to break this link, you must change the inserted picture to the **Static Picture** type using the Property List (see the following paragraph).
  The properties of static pictures and library pictures are generally identical.

3 Reposition the picture as desired and set its properties.

The picture has a set of object properties, just like any object on the form. If you like, you can modify these properties.

Dissociating a Picture from its Library Source

When you insert a picture that comes from the Picture Library, you actually insert a reference to a picture. The Property List will indicate its Type (under Objects) as a Library Picture. If the picture is modified in the Picture Library, each instance of it will be modified accordingly.

You may want to dissociate a picture inserted in a form from its source in the Picture Library.

► To disassociate a picture from its source in the Picture Library.

1 Display the picture's properties in the Property List.

2 In the "Objects" theme, click the Type line.

Two types are available: **Library Picture** (default value) and **Static Picture**.

3 Select Static Picture.

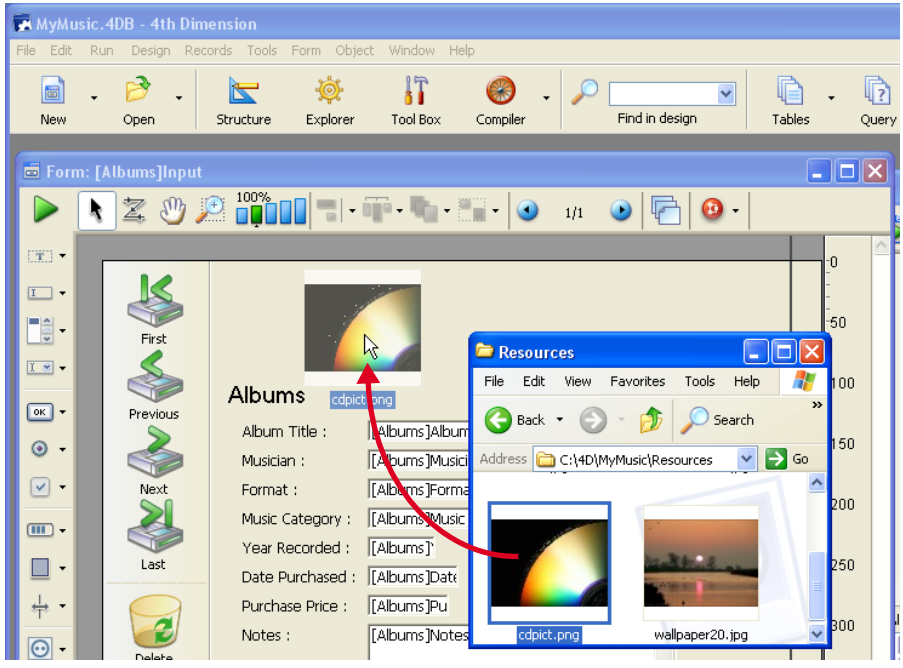
The picture is then treated as if it had been pasted from the Clipboard or from a dropped file.

Automatic Referencing of Picture Files

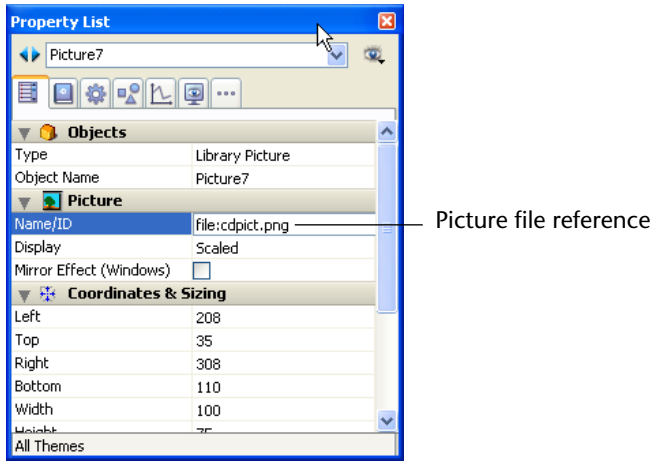
You can use the **Resources** folder for storing static pictures used in your forms and work with them by reference. The display of these pictures is then optimized and their management is greatly facilitated.

Note For more information about the Resources folder, please refer to the "Database Architecture" paragraph on page 42.

More particularly, you can call on .png (bitmap) or .svg (vectorial) pictures. Inserting these pictures into forms can be carried out by a simple drag and drop operation from the **Resources** folder:



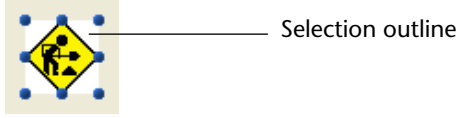
4D automatically inserts the picture reference into the form in the following manner: “file:{*pathname+*}filename”:



It is possible to place pictures in subfolders of the Resources folder. In particular, you can use the *lproj* folder mechanism for pictures in different languages (for more information about this mechanism, please refer to appendix C “XLIFF Architecture”on page 1293).

Modifying the Background of a Picture

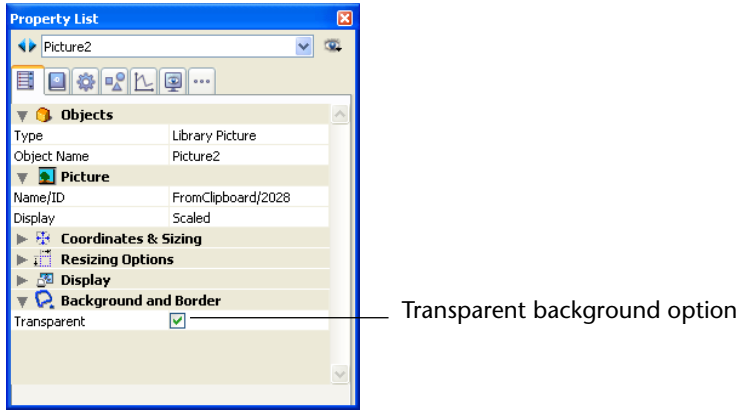
By default, pictures are inserted with a (white) selection outline. You can change the picture’s background to Transparent so that it takes the form’s background color



► To modify the picture’s background:

1 Double-click the picture to display its properties in the Property List.

2 **Expand the "Background and Border" theme and check the Transparent option.**



The background of the picture becomes transparent:

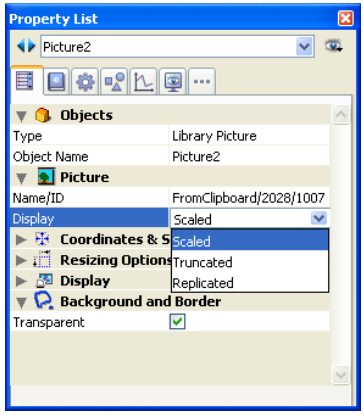
Setting the Display Mode for a Static Picture

You can set the display mode of a static or library picture that is placed in a form. The display mode determines the way a picture must be redrawn when the area containing it is resized.

► To set the display mode for a picture:

1 **Double-click the picture to display its properties in the Property List.**

2 **In the "Picture" theme, choose an option from the Display menu.**



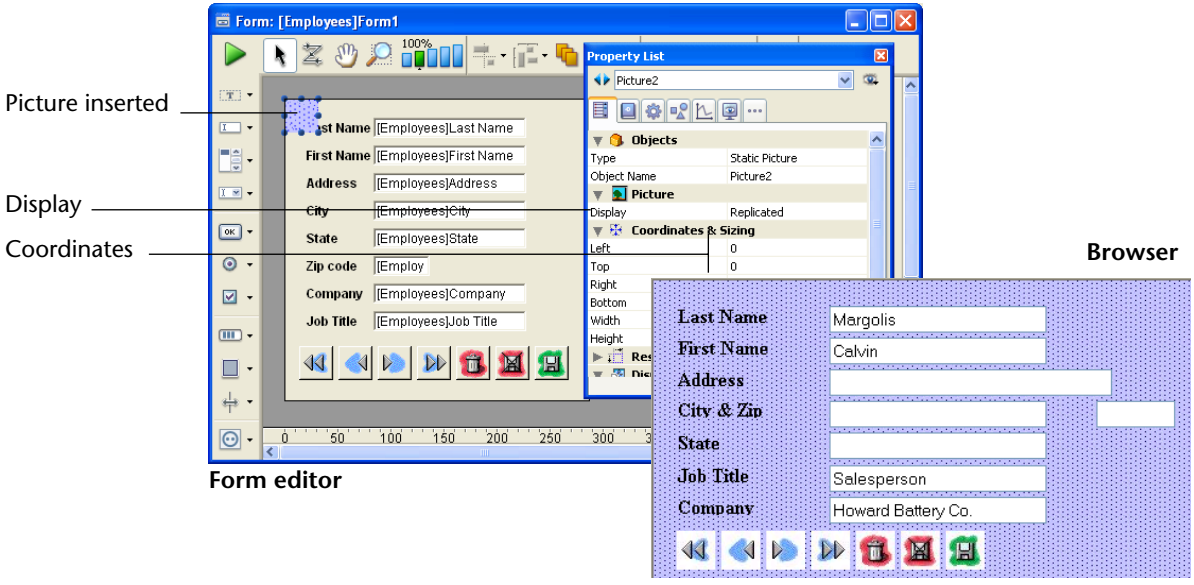
The following options are available:

- **Scaled (default mode)**: When the picture object is resized, the picture is resized so that the entire picture remains visible.
- **Truncated**: When the picture object is resized, the picture keeps its proportions and only its boundaries change. The picture always stays in the center of the picture object. If the picture object is reduced to a smaller size than the picture, the picture is truncated.
- **Replicated**: When the size of the picture object is increased, the picture is replicated as many times as necessary to fill the new area. Recommended for background pictures since it doesn't require a lot of memory. The Form Wizard uses this option when selecting the **Background Picture** option (please refer to "Display Options" on page 358). If the picture size is reduced to a smaller size than the original picture size, the picture is truncated (not centered).

Defining a Background Picture for Web Forms

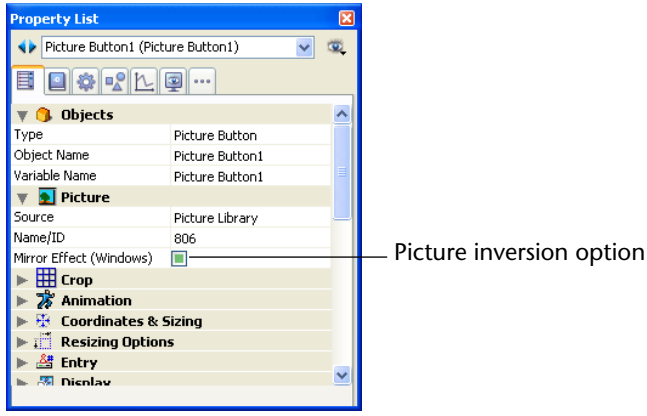
You can insert a static picture in a form designed to be published on the Web and use that picture as a background replicated picture for browsers. To do so, the picture settings must meet the following conditions:

- The picture must be located in the upper left corner of the form (coordinates (0,0,x,x).
- The Replicated option has to be assigned to the Display property of the picture.



Mirror Effect for Pictures (Windows)

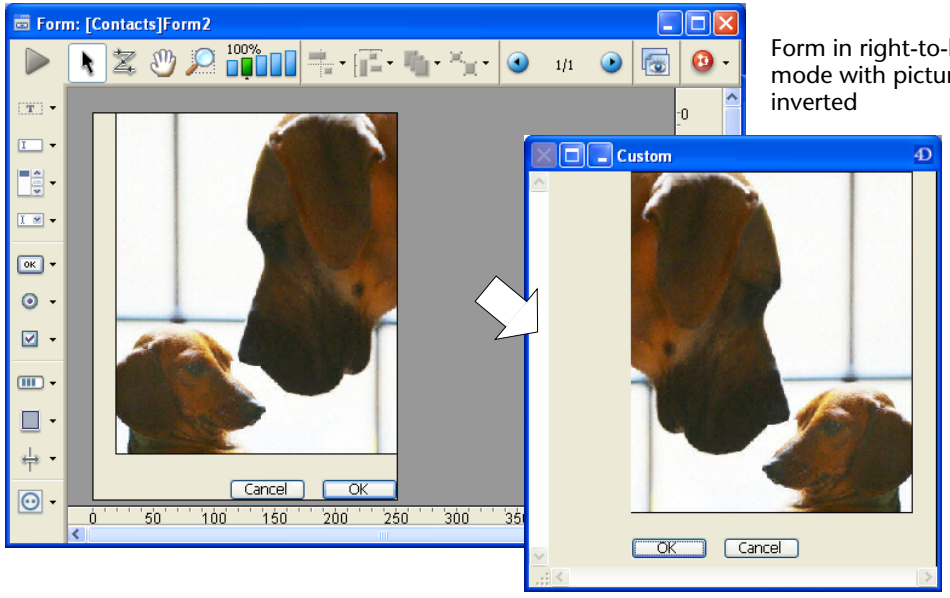
The **Mirror Effect (Windows)** option is found in the "Picture" theme of the Property List for pictures displayed in forms:



This option can be used in applications configured in "right-to-left" languages that have an inverted version of the interface (for more information about "right-to-left" mode, please refer to the "Right-to-left Languages" paragraph on page 201).

By default, the right-to-left mode does not reverse the contents of pictures in forms. However, it may sometimes be necessary to invert certain pictures. This is the case, for example, with pictures used in the interface, such as arrows, that are linked with the position of objects.

When the **Mirror Effect (Windows)** option is checked, a left-to-right (mirror effect) inversion is applied to the picture:



Form in right-to-left mode with picture inverted

This option is available for static pictures and library pictures, as well as for active objects using pictures (see the "Mirror Effect (Windows)" paragraph on page 545.

This option is only taken into account:

- In Application mode,
- Under Windows,
- When the "right-to-left" mode is enabled for the form.

Using Object Libraries

4D allows creating and using *object libraries*.

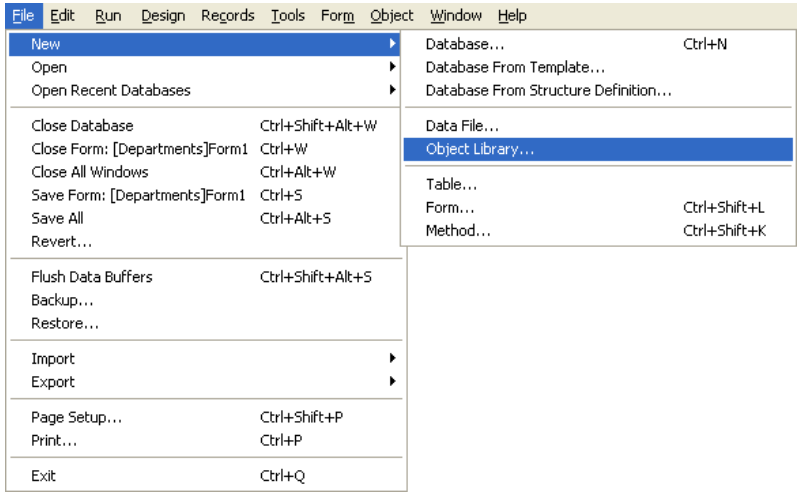
An object library is an external file created by 4D. It allows storing all types of objects that can be used in forms: buttons, texts, pictures, hierarchical lists, etc. Objects are stored with all their properties, including their object methods. Libraries are put together and used by a simple drag-and-drop or copy-paste operation. They are somewhat like a permanent clipboard.

Using libraries, you can build form object backgrounds grouped by graphic families, by behavior, etc. Since these libraries are stored as an external file, their use with different databases is quite easy.

Finally, you can combine object libraries with user forms in order to supply objects that can be used to customize databases. These principles are described in the chapter “User Forms” on page 685 and in the *Language Reference* manual of 4D.

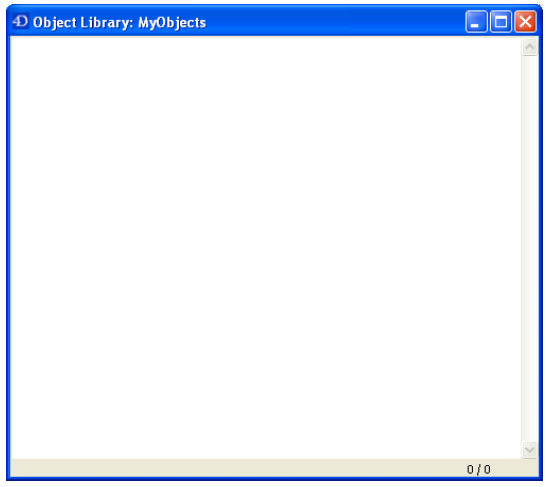
Creating an Object Library

To create an object library, select the **New>Object Library...** command from the 4D **File** menu or tool bar.



A standard save file dialog box appears, which allows you to choose the name and the location of the object library. The extension of libraries is “**.4il**.”

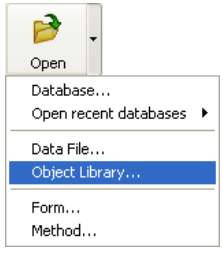
Once you validate the dialog box, 4D creates a new object library on your disk and displays its window (empty by default):



You can create as many libraries as desired per database. A library created and built under Mac OS can be used under Windows and vice-versa.

Opening an Object Library

To open an existing object library, select the **Open>Object Library...** command in the 4D **File** menu or tool bar:



A standard open file dialog box appears, which allows you to select the object library to open.

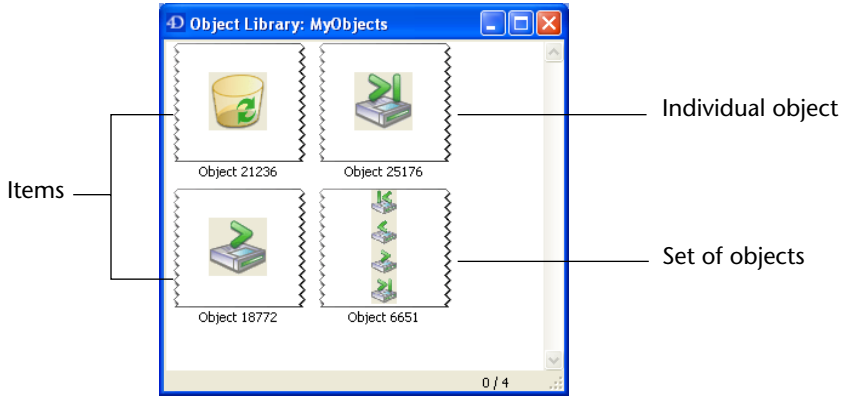
A given object library can only be opened by one database at a time. However, several different libraries can be opened in the same database.

Building an Object Library

Objects are placed in an object library using drag-and-drop or a cut-copy-paste operation. They can come from either a form or another object library. No link is kept with the original object: if the original is modified, the copied object is not affected.

Note In order to be able to drag and drop objects from forms to object libraries, you must select the "Start drag and drop" option in the 4D Preferences (see the "Move" paragraph on page 181).

You can place individual objects or sets of objects in an object library. Each object or set is grouped into a single item:

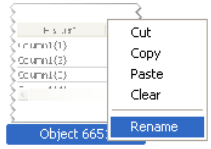


An object library can contain up to 32,000 items.

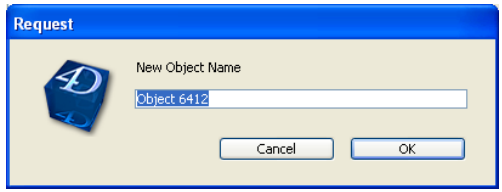
Objects are copied with all their properties, both graphic and functional. These properties are kept in full when the item is copied into a form or another library.

Each new item is named "Object," followed by a random number, for example *Object2012*. This item can be renamed as desired. To do that, you can:

- Double-click the item to rename
- **Right-click** the item to rename, then choose the **Rename** command in the contextual menu:



A dialog box appears allowing you to rename the item:



More than one item can have the same name.

Contextual Menu

Each object library has a contextual menu that allows accessing basic management commands. The contents of this menu depend on where the mouse click occurs:

Click outside an item

Click on an item

Note It is impossible to select several items simultaneously.

- **Cut / Copy**: These commands allow placing a selected item of the object library in memory. The **Cut** command removes the item from the library. In both cases, the item can then be pasted into a form or into another object library.
- **Paste**: This command is only active if a library-compatible object is found in memory. It allows inserting the object in memory into the object library.

Note These commands are also available using the **Edit** menu and standard keyboard shortcuts.

- **Clear**: This command removes the selected item from the library. You can also use the **Del** or **Backspace** keys.
- **Rename**: This command displays the dialog box for renaming the item (see above).

Dependent Objects

Using copy-paste or drag-and-drop with certain library objects also causes their dependent objects to be copied. For example, copying a button will cause the object method that may be attached to be copied as well. These dependent objects cannot be copied or dragged and dropped directly.

The following is a list of dependent objects that will be pasted into the library at the same time as the main object that uses them (when applicable):

- Lists
- Stylesheets
- Formats/Filters
- Pictures
- Help Tips (linked to a field)
- STR# resources
- Object methods.

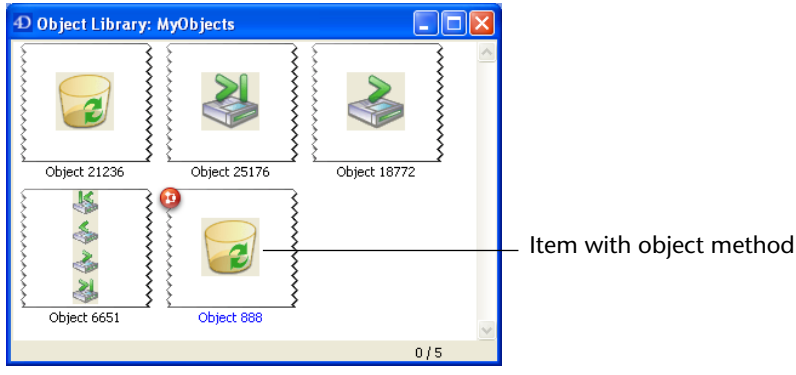
For STR# resources, the entire STR# is copied and not just the string associated with the object.

Displaying Object Libraries

Object libraries can be accessed from the 4D Design environment and, if the developer allows it, in the User form editor in Application mode (see the chapter “User Forms” on page 685 and the *Language Reference* manual).

All library items are visible in the Design environment. However, only items without an associated object method are visible in the User form editor.

Items with at least one associated object method are displayed with a shield (the same one displayed on objects with an object method in a form, see the “Using Shields” paragraph on page 434) and their name appears in blue:



This allows easily distinguishing items that are visible for the developer in the Design environment from those that are visible for the user in the Application environment.

Client-Server Behavior

In order for all client machines to be able to access an object library, the library file must be placed in a folder named **Extras**, which is placed at the same level as the database structure opened by 4D Server. The Extras folder can be used for transferring custom items of the server to the client machines (resources file, text documents, XML preferences files, etc.).

When a 4D Client connects to the server, the Extras folder is moved to the client machine. 4D Server automatically handles any modifications made to this folder and only transfers what is necessary. Moreover, the contents of the folder is compressed in order to optimize network copying time.
On the client side, the **Extras** folder is downloaded to the same location as the other structure elements, i.e.:

System	Location of downloaded folders on the client machine
Mac OS	{Disk}:Library:Application Support:4D:*DatabaseName_Address*:Extras
Windows	{Disk}:\Documents and Settings*User Name*\Application Data\4D*DatabaseName_Address*\Extras

This is valid regardless of the server and client platform.

Creating a Multi-page Form

You can create multiple pages for an input form. If you have more fields than will fit on one screen, you may want to create additional pages to display them. Multiple pages allow you to do the following:

- Place the most important information on the first page and less important information on other pages,
- Organize each topic on its own page,

- Reduce or eliminate scrolling during data entry,
- Provide space around the form elements for an attractive screen design.

Note Multiple pages are a convenience used for input forms only. They are not for printed output. When a multi-page form is printed, only the first page is printed.

There are no restrictions on the number of pages a form can have. The same field can appear any number of times in a form and on as many pages as you want. However, the more pages you have in a form, the longer it will take to display it.

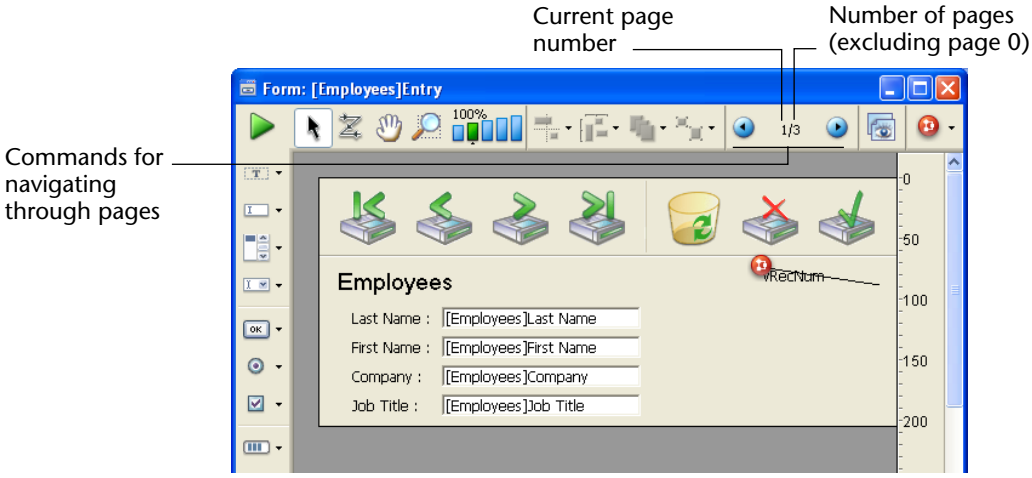
A multi-page form has both a background page and several display pages. In the Form editor, objects that are placed on the background page may be visible on all display pages, but can be selected and edited[1] only on the background page. In multi-page forms, you should put your button palette on the background page. You also need to include one or more objects on the background page that provide page navigation tools for the user. For information on adding page navigation tools, see the section "Adding Page Navigation Controls" on page 471. This section details how to add and delete pages, add objects to the background page, move from page to page, and add fields to a new page.

Adding a Display Page to a Form

Every form has at least one display page[2] and a background page. The current page number appears in the toolbar of the form window. This area also includes a pop-up menu that allows you to select the displayed page.

1. When the form is executed, objects placed on the background page are always visible on all the pages.
2. The Options page of the Form Wizard contains an option that instructs the Form Wizard to create a multi-page form automatically if the fields you select don't fit on one page. If you selected this option, your form may initially have more than one display page. When the Form Wizard creates more than one display page, it puts buttons, variables, the form title, and decorative rectangles on the background page.

The background page is numbered zero (0).



To add a display page:

1 **Choose Add Page from the Form menu.**

4D creates a new page.

OR

Move to the last page of the form, then click the Next Page icon in the toolbar.

4D displays a dialog box asking if you want to add a page. Click **OK** to create the new page.

A new, blank display page appears in the Form editor window. The page number area of the window displays the number of the page you are viewing.

You can now add fields and other form elements to the new page.

► To insert a display page before the current page:

1 **Choose Insert Page from the Form menu.**

4D inserts a new page before the current page and displays it on screen.

Moving from Page to Page

When you want to display the background page or move to another display page, you can either use the page navigation tools in the Tools palette or the page pop-up menu in the Form editor window.

► To display the background page (page 0):

■ **Move to the first page of the form and click the Previous Page icon in the toolbar.**
OR
Use the Page pop-up menu to move to page 0.
OR
Select 0 from the Goto Page submenu in the Form menu
OR
Press Alt+click (Windows) or Option+click (Mac OS) on an object belonging to page 0, or specifically outside any object belonging to the current page[1].

4D displays the background page. The page number of the background page is zero (0). Objects located on the background page are displayed on each page. You can place any object type on the background page.

Note There should be no confusion between using the **Page 0** menu item from the **Display** submenu and actually making the background page the currently displayed page. Selecting the **Page 0** menu item from the **Display** submenu only displays or hides the items of the background page *in the current page*. For more information, refer to "Showing/Hiding Elements in the Form Editor" on page 374.

► To use the page navigation tools:

■ **To move to the next page, click the Next Page icon in the toolbar.**

■ **To move to the previous page, click the Previous Page icon in the toolbar.**

4D displays the page immediately following or prior to the current page.

1. The **Alt+click** or **Option+click** shortcuts on an object of the current page creates or opens the object method.

If you click **Previous Page** while viewing the first page of the form, the background page appears. If you click **Previous Page** while viewing the background page, nothing happens. If you click **Next Page** while viewing the last page of the form, 4D asks if you want to create another page for the form.

► To display any page:

1 **Click on the page number area and hold down the mouse button:**

OR
Display the Goto Page submenu of the Form menu.

2 **Choose the desired page number.**

Deleting a Page

You can delete unwanted display pages from a multi-page form. Any fields or other objects on the deleted pages will be deleted as well. The remaining pages are renumbered. You cannot delete the first page or the background page in a form that consists only of these two pages.

► To delete a page from the form:

1 **Use either the page navigation tools or the page pop-up menu to display the page you want to delete.**

2 **Choose Delete Page from the Form menu.**
A dialog box appears verifying that you want to delete the page from the form.

3 **Click OK.**
The page and any objects on the page are removed from the form.

Adding Page Navigation Controls

When you create a multi-page form, you need to provide a way for users to move from one page to another. 4D provides three ways that you can use to add navigation tools:

- **Tab control**: The tab control object gives users random access to individual pages. You place the tab control on the background page of the form and use its properties to provide page navigation controls.
- **Automatic buttons**: You can add automatic page navigation buttons to the form — First Page, Last Page, Previous Page, and Next Page. These buttons should be placed on the background page.

- **Object methods**: In addition, the language includes the GOTO PAGE command. You can use this command as part of an object method to create custom navigation controls using any suitable object type. For example, you can choose to use a picture button or pop-up menu to provide page navigation controls.

Adding Page Navigation Buttons

You can include page navigation buttons when you generate the form using the Buttons page of the Form Wizard. After the form is generated, open it in the Form editor and add the necessary pages. If you need to add the page navigation buttons after the form is created, you can do so using the button creation tool in the object bar. For more information, see "Creating an Active Object" on page 540.

Using a Tab Control

The tab control provides a visual indication of the current page and the remaining pages. For information on creating and activating a tab control, see the section "Tab Controls" on page 585.

Inherited Forms

4D allows you to use "inherited forms." The principle behind this feature is to be able to use objects from Form A in a Form B: Form B "inherits" the objects from Form A.

Suppose, for example, that all entry forms belonging to a database have to contain the **OK**, **Cancel**, **Next**, and **Previous** buttons as well as a logo. Simply create a form containing only these elements and then call it as an inherited form in all database entry forms. Each entry form contains only fields and objects specific to its use.

Unlike form templates set using the Form Wizard (see the "Creating a Form Template" paragraph on page 366), the reference to the inherited form is always active: if an element of the inherited form is modified (button styles, for example), all forms using this element will automatically be modified.

Using Inherited Forms

Once using the database, inherited form objects are dynamically combined with those of the open form. This mechanism is very similar to that of the "page 0" form mechanisms, the difference being that it can be applied to all of the database forms as a whole.

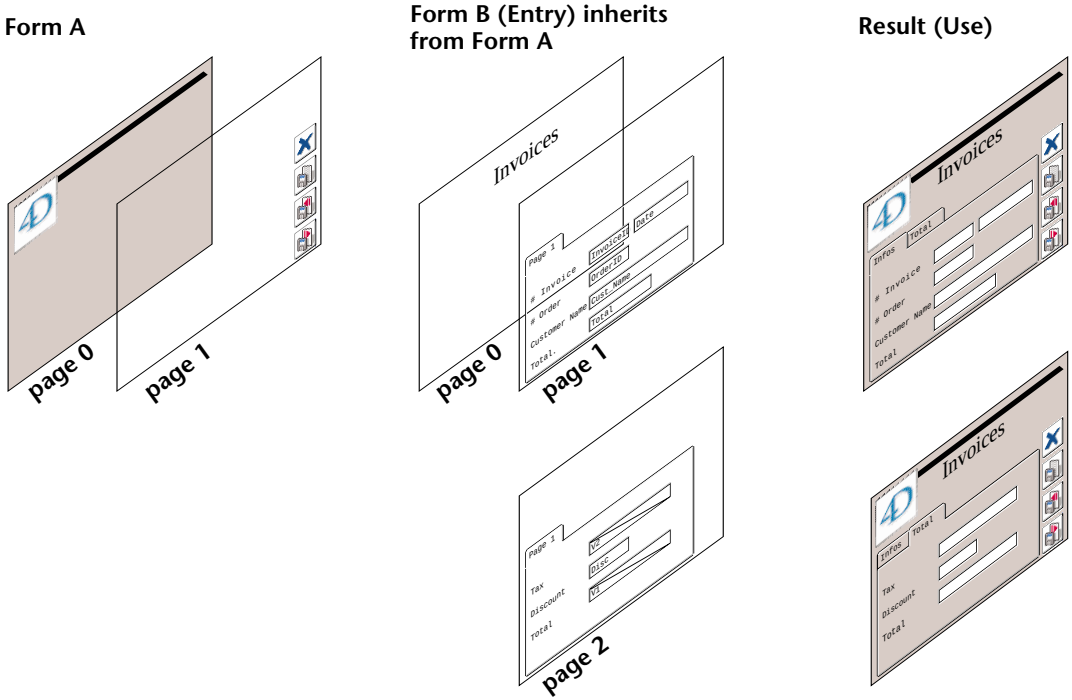
When the form is executed, the objects are loaded and combined in the following order:
1- Page zero of the inherited form
2- Page 1 of the inherited form
3- Page zero of the open form
4- Current page of the open form

This order determines the entry order of objects in the form.

Note Only pages 0 and 1 of the inherited form can appear in other forms.

The properties (window name, re-sizing, events, etc.) and the method of an inherited form are not considered when used as an inherited form. On the other hand, the methods of objects that it contains are called.

The following diagram illustrates how inherited forms work



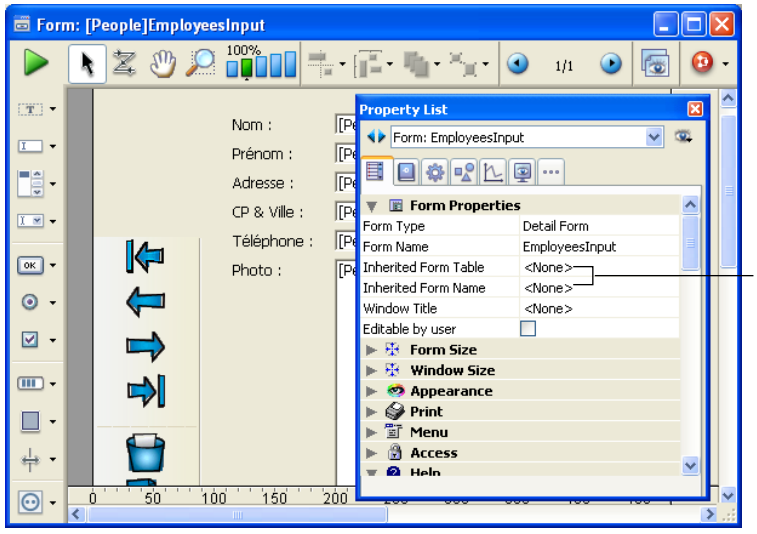
Defining an Inherited Form

Form inheritance starts in the 4D Form editor.

► To define an inherited form:

1 In the Form editor, open the form before inheriting another form.

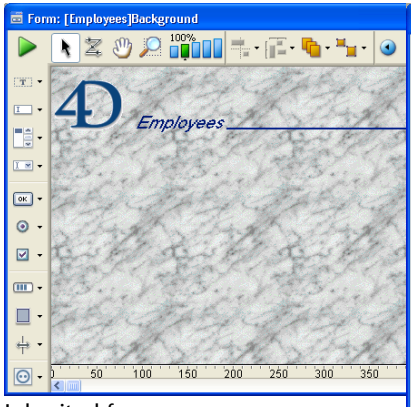
2 Display the Property List and click outside all objects in the form in order to see form properties.



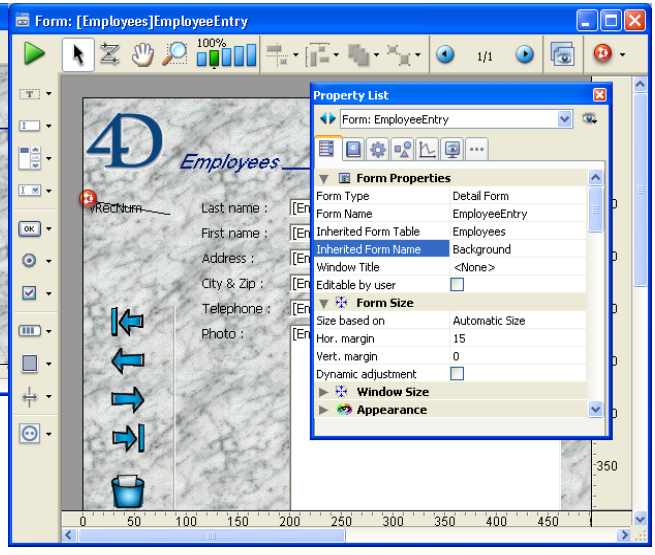
The “Inherited Form Table” and “Inherited Form Name” lists are available. All database tables and their forms are displayed.

3 Select the table and then the form name to inherit.

All forms can be designated as inherited form. However, the contained elements must be compatible with use in different database tables.



Inherited form (example)



Final version

Once an inherited form is selected, its contents appear in the current edit window. This is only a preview; it is not possible to select or modify an object in this form. To do that, you should open it in its own window.

Note You can hide the objects of an inherited form by deselecting the **Inherited form** option in the **Display** submenu of the **Form** menu or in the editor's contextual menu.

To stop inheriting a form, select the <**None**> option in the Property List.

Note It is possible to define an inherited form in a form that will eventually be used as an inherited form for a third form. The combining of objects takes place in a recursive manner. 4D detects recursive loops (for example, if form [table1]form1 is defined as the inherited form of [table1]form1, in other words, itself) and interrupts the form chain.

Data Entry Order

The data entry order is the order in which fields, subforms, and other active objects are selected as you hit the **Tab** or the **Carriage return** key in an input form. It is possible to move through the form in the opposite direction (reverse data entry order) by pressing the **Shift**+**Tab** or **Shift**+**Carriage return** keys.

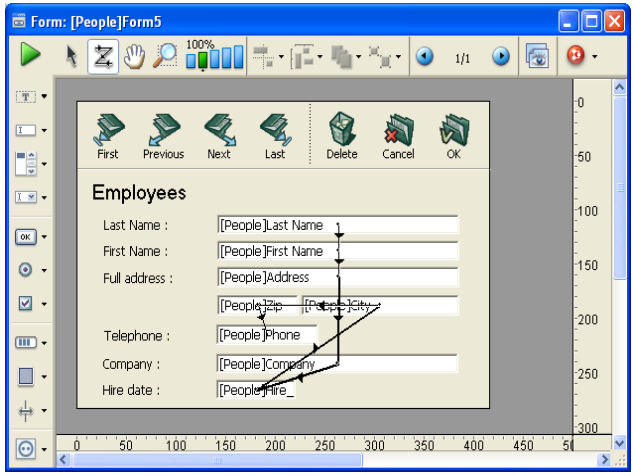
Every enterable area into which you can type a value is included in the data entry order. Boolean fields (shown as radio buttons or check boxes), subforms, combo boxes, and areas that accept pictures are also included in the data entry order.

Thermometers, rulers, and dials can also be used to enter data. These objects, however, are not included in the data entry order. You select them by clicking on them.

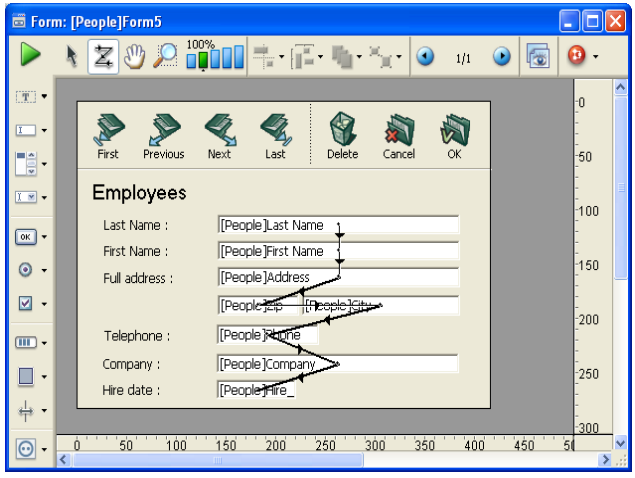
If you don't specify a custom entry order, by default 4D uses the layering of the objects to determine the entry order in the direction "background towards foreground." The standard entry order thus corresponds to the order in which the objects were created in the form.

In some forms, a custom data entry order is needed. Below, for example, additional fields related to the address have been added after the creation of the form.

The resulting standard entry order thus becomes illogical and forces the user to enter the information in an awkward manner.



In cases such as this, a custom data entry order allows you to enter the information in a more logical order.



Viewing and Changing the Data Entry Order

You can view the current entry order either using the “Entry order” shields, or by using the “Entry order” mode.
However, you can only modify the entry order using the “Entry order” mode.

This paragraph describes viewing and modifying the entry order using the “Entry order” mode. For more information about viewing the entry order using shields, refer to the “Using Shields” paragraph on page 434.

To switch to “Entry order” mode, use the **Entry Order** command of the **Form** menu or the corresponding button of the toolbar.

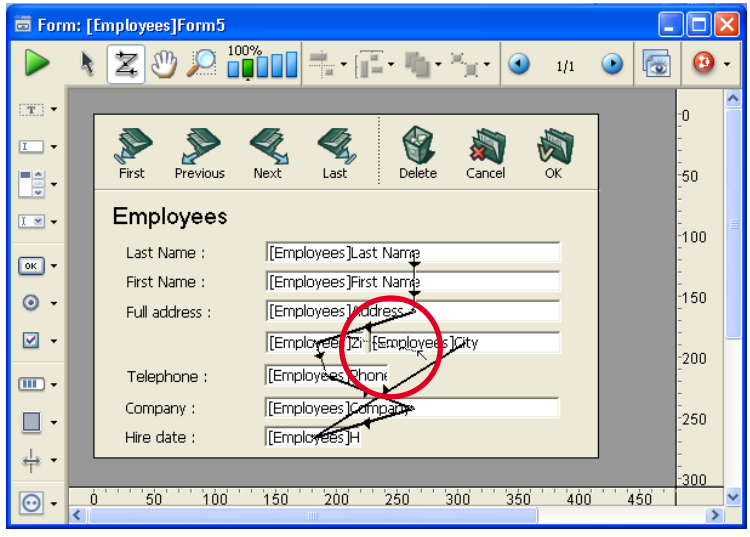
► To view or change the entry order:

1 **Choose Entry Order from the Form menu or click on the button in the toolbar of the window.**

The pointer turns into an entry order pointer , and 4D draws a line in the form showing the order in which it selects objects during data entry.

Viewing and changing the data entry order are the only actions you can perform until you click any tool in the Tools palette.

2 **To change the data entry order, position the pointer on an object in the form and, while holding down the mouse button, drag the pointer to the object you want next in the data entry order.**



4D will adjust the entry order accordingly.

3 **Repeat step 2 as many times as necessary to set the data entry order you want.**

4 **When you are satisfied with the data entry order, click any unselected tool in the toolbar or choose Entry Order from the Form menu.**

4D returns to normal operation of the Form editor.

Note Only the entry order of the current page of the form is displayed. If the form contains enterable objects on page 0 or coming from an inherited form, the default entry order is as follows: Objects from page 0 of the inherited form ➔ Objects from page 1 of the inherited form ➔ Objects from page 0 of the open form ➔ Objects from the current page of the open form.

Setting the First Object in the Data Entry Order

All enterable objects are part of the data entry order. To set the first object of the entry order, you must modify its location among the form levels. The Entry order mode must be disabled.

► To establish one of the objects as the first in the data entry order:

1 **Select the object you want to be first in the entry order.**

2 **Choose Move to Back from the Object menu.**
OR
Select Move to Back from the Level submenu of the contextual menu for the object.
The object will be placed in back of any other form element.

3 **Choose Entry Order from the Form menu or click the**  **button in the toolbar.**
The selected object becomes the first object in the entry order and the object that was first becomes second. You can now drag from object to object in order to create the data entry order you want.

4 **When you have finished, select Entry Order from the Form menu again or click any unselected tool in the toolbar.**
The Form editor returns to normal operation.

Using a Data Entry Group

While you are changing the data entry order, you can select a group of objects in a form so that the standard data entry order applies to the objects within the group. This allows you to easily set the data entry order on forms in which fields are separated into groups or columns.

► To create a data entry group:

1 **Choose Entry Order from the Form menu or click the**  **button in the toolbar.**

2 **Draw a marquee around the objects you want to group for data entry.**

When you release the mouse button, the objects enclosed or touched by the rectangle follow the standard data entry order. The data entry order for the remaining objects adjusts as necessary.

Restoring the Standard Data Entry Order

You can restore the standard data entry order (based on the layering of the objects) at any time.

1. **Choose Entry Order from the Form menu or click the  button in the toolbar.**
2. **Draw a marquee around all the objects on the form.**
   When you release the mouse button, the objects enclosed or touched by the rectangle take the standard data entry order.

Excluding a Field from the Entry Order

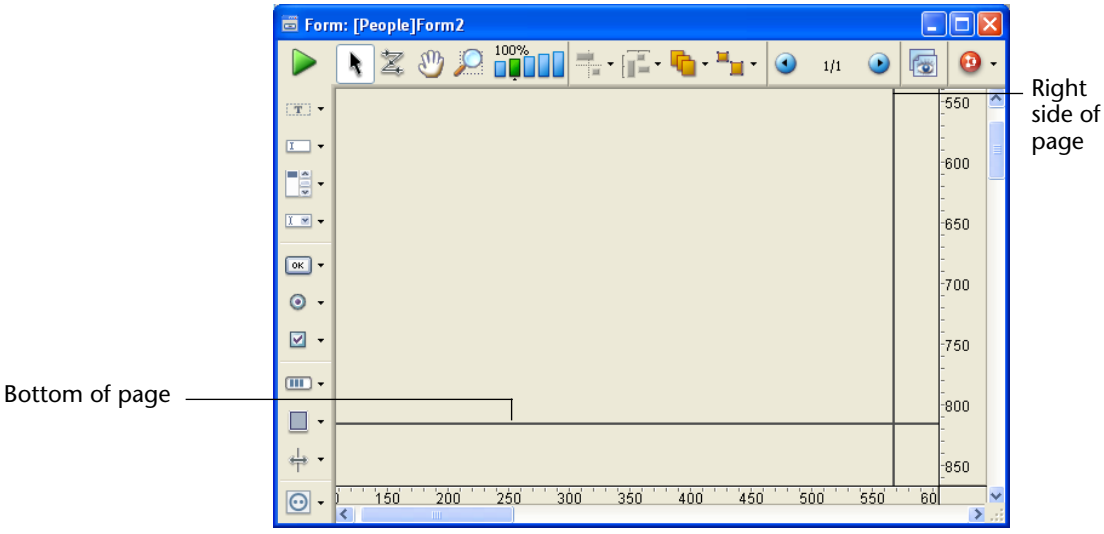
By default, all the fields are included in the entry order. If you want to exclude a field from the entry order while keeping its "Enterable" property, you simply need to deselect the **Tabable** property for this field. For more information, refer to the "The Tabable Attribute" paragraph on page 491.

Viewing and Printing Forms

Each form has a maximum area of about 1245 square feet. You scroll to bring hidden portions of the form into view. For viewing on screen, your form design can use this entire area. You can scroll to view any element you place in the form.

For printing, form elements must fit within a single page width, but may be several pages in length. The actual size of a page depends on your printing device, the paper it is using, and the specifications you enter in the Page Setup dialog box. 4D displays page border lines in the Form editor. These lines indicate the page limits. The page border lines respond to any page setup changes. The page setup specifications are stored with the form when it is closed. The form's limits can be displayed or hidden using the **Paper** command in the **Display** submenu of the Form editor (see "Showing/Hiding Elements in the Form Editor" on page 374).

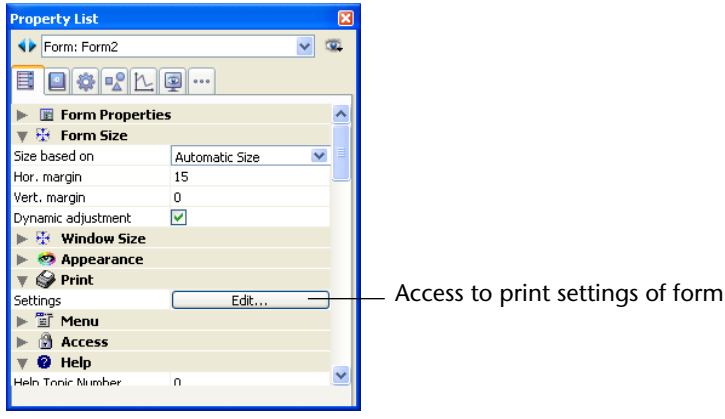
The figure below shows the page border lines:



Form Print Settings

You can set specific print settings for each form. These settings will be taken into account when the form is printed in Application mode.

To do this, click on the Settings/**Edit...** button in the "Print" theme of the Property List of the form. A print setup dialog box appears, which lets you modify the specific print settings of the form:



Note These settings are different from the current print settings that can be accessed from the **File** menu. The latter are used for the current session in the Design mode and are not stored.

Saving Forms

It is a good idea to save any changes you make to a form, especially when using 4D Server with multiple users. You can save a form by closing or saving it. You can close a form by clicking its close box or by choosing **Close Form:** ***Name*** from the **File** menu.

To save a form without closing it, choose **Save Form:** ***Name*** from the **File** menu.

Once a form has been saved, you can continue to work on it. If you make a mistake or do not like the changes you have made, you can revert to the last saved version of the form. This makes the form appear exactly as it did the last time that it was saved. To do this, choose **Revert...** from the **File** menu.

4D Server When a form is saved in the Design environment, users are able to see your changes the next time they open the form.

7 Fields and Active Objects in Forms

4D allows you to customize data entry forms so that your interface includes exactly those features that you need. You can add interface elements such as picture buttons, tab controls, drop-down lists, combo boxes, list boxes and hierarchical lists to your data entry forms. You can also implement drag-and-drop operations. You can use triggers, form methods, or object methods to enforce business rules during data entry.

This chapter tells you how to do the following:

- Place fields and other active objects in the form,
- Set display formats and data entry filters,
- Use data entry constraints such as maximum, minimum, default, or required values,
- Write form or object methods,
- Add interface objects such as buttons, pop-up menus or drop-down lists, combo boxes, scrollable areas, list boxes, splitters and tab controls,
- Add subforms to forms,
- Attach custom menus to forms.

Active Objects Defined

An active object is anything on a form that performs a database task or an interface function. There are many kinds of active objects. Fields are considered active objects. Other active objects — enterable objects (variables), combo boxes, drop-down lists, picture buttons, and so on — store data temporarily in memory or perform some action such as opening a dialog box, printing a report, or starting a background process.

In some cases, you can specify the active object's action by making selections in the Property List window. For example, you can use built-in automatic button actions to specify the action of a button. In other cases, you specify the object's action by writing a method that is automatically attached to the object.

There are also instances in which you will want to manage active objects at a higher level. For example, data validation tasks can be handled by a trigger that runs automatically when a record is saved. The trigger can examine the values in each field for possible violations of business rules.

Fields in a Form

The fields of a form are used to enter or display the data of a record.

When you create a new form using the Form Wizard, you select the fields that you want to include in the form. Once the form has been created, you can use the Form editor to specify additional properties such as the display format and data entry controls.

These properties only apply to the forms in which they were specified. You can then use the same properties for other forms or specify new ones. You can change the properties of the fields or add/delete them at any time.

Adding Fields to a Form

You can add or delete fields from a form at any time. For example, you may decide to add fields to a form when the following occurs:

- You discover you need a field you did not choose in the Form Wizard.
- You add a field to the database structure and need to add it to a form so that you can use it.

When you place a field in a form, you can immediately set its properties. You can add a field to a form using either the toolbar or the Explorer.

▶ To add a field to a form using the toolbar:

1 **If the field does not exist in the database, use the Structure editor to create the field.**

For information on adding a field to a table, see the section "Creating Fields and Setting Field Properties" on page 265.

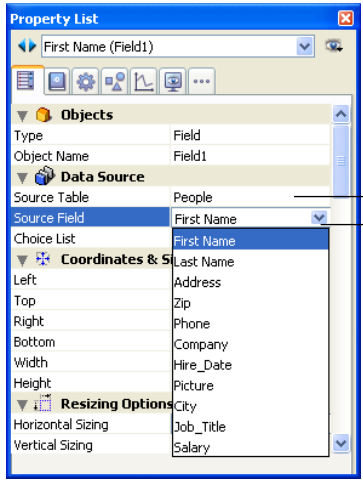
2 **Open the form to which you want to add the field.**

For information on opening a form, see the section "Opening a Form in the Form Editor" on page 392.

3 **Select the field insertion tool in the toolbar then draw the field area in the form.**

4D automatically displays the properties of the new field in the Property List.

4 **Select the field you want to insert from the table/field list.**



5 **If desired, select the specific properties you want to assign to the field.**

After creating the field, you generally need to set additional properties. You can set data entry controls, write help text, attach a method, set resizing or repositioning options, set font or appearance options. The new field appears in the form where you placed it. The field area displays the name of the field you selected, preceded by the table name.

For information on field attributes, see the sections "Setting Field Properties" on page 277 and "Setting the Enterable and Mandatory Attributes" on page 490.
For more information about field properties, please refer to "Modifying a Field in a Form" on page 486.

► To add a field using the Explorer:

1 **Open the form to which you want to add the field.**

For information on opening a form, see the section "Opening a Form in the Form Editor" on page 392.

2 **Open the Explorer and click the Tables button to display the hierarchical list of tables and fields.**

3 **Expand the table that contains the field you wish to add.**

The fields belonging to that table appear.

4 **Drag the desired field from the Explorer to the form.**

4D displays the field properties in the Property List.

After you place a field in a form, you can modify it as you would any other form object. You can resize it, change the font, choose colors for display on a color monitor, and so on.

Modifying a Field in a Form

As for any active object, you can modify the field properties in a form at any time. Numerous options can be used to control the appearance, data entry or display of the fields.

► To modify a field's properties:

1 **If the Property List is already displayed, click on the field that you want to modify; otherwise, double-click it.**

The properties are displayed in the Property List.

You can use the Property List to change any setting. You can even select a new field in the form to display its properties and you can select a set of fields to modify their common properties.

2 **Make any changes you like.**
Your changes take effect immediately.

Note If the field has been grouped with another object, in principle you must ungroup it before you can display its properties. However, you can also use the **Ctrl+click** (Windows) or **Command+click** (Mac OS) shortcut on a grouped object in order to select it without having to ungroup it.

Changing a Field into a Variable and Vice-Versa

You can transform every object type (active or not) into another object type. You can also transform a field into a variable and vice-versa. This is useful when, after inserting a field in a form, you want to change that field into a variable because you don't need to store the value. When 4D changes an object into another object, it keeps the original properties of the object (coordinates, object method, appearance, color and so on).

The data type assigned to a field will be kept for the variable: a picture field will be converted to a picture variable.

To change a variable into a field or a field into a variable, select the object and select Field or Variable from the **Type** drop-down list in the Property List ("Objects" theme). The Property List is then updated to display the properties for that new object type. The object name, object method, and its properties (size, enterable and so on) remain identical.

When you change a variable into a field, 4D assigns the first field in the first table to the object by default. You can manually set the table and field in Source table and Source field ("Data Source" theme).

Inserting Dynamic Table and Field Names

You can insert dynamic table and field names into your forms (as static text). When you place dynamic table or field names in your forms, they are automatically updated throughout your database:

- Either when you modify the table or field name in the Structure editor, or
- When the 4D commands 4D SET TABLE TITLES and SET FIELD TITLES ("User Interface" theme) are called.

This feature is particularly useful when you want to update the table or field names in the forms on the fly.

Note Dynamic references can also be inserted into static text areas (see the "Working with Static Text Areas" paragraph on page 445), into form window titles (see the "Default Window Title" paragraph on page 401) and into help tips (see the "Adding Help Tips to a Field or Object" paragraph on page 517).

► To insert a dynamic table or field name in a form:

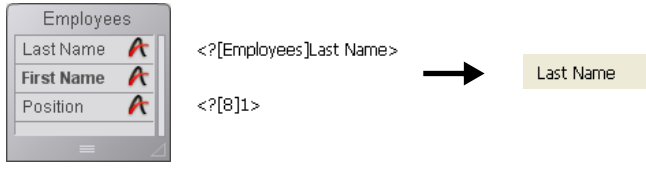
1 In a static text area, enter the following reference:

- To insert a dynamic table name: **<?[TableName]>** or **<?[2]>** (the table's creation order number, meaning the second table created).
- To insert a dynamic field name: **<?[TableName]FieldName>** or **<?[2]3>** (the table's and field's creation order number), or even **<?3>** (the field's creation order number) for a field of the current table.

Warning: The number of fields and tables that have been deleted can be reused by 4D.

2 Click outside of the text area.

The current field's or table's name appears as it has been defined in the Structure editor.



You can view the "actual" contents of a static area in the Form editor by clicking on the area or by selecting **Show Format** or **Show Name** from the **Object** menu.

In the Application environment, a table's or field's name can be modified on the fly by using the SET TABLE TITLES and SET FIELD TITLES commands. In this case, the table and field name references will display the values set by these commands.

Notes •Dynamic field names are available as an option on the **Options** page of the Form Wizard. For more information, refer to "Display Options" on page 358.

• Since table and field titles are text areas, they can also be specified using XLIFF or STR# references (see Appendix C on page 1293).

Data Entry Controls

You can establish data entry controls for fields and enterable objects at the form level. Data entry controls restrict what the user can enter into the field or enterable object on a particular form. You can do the following:

- Set the Enterable or Mandatory attributes,
- Attach a choice list,
- Establish a list of required or excluded values,
- Set an entry filter that defines allowable characters,
- Set maximum and minimum allowable values,
- Set default values,
- Write an object method.

You can also establish data entry controls at a higher level, i.e., in the database structure. You have the following options:

- **Field properties**: The Inspector window (accessed from the Structure editor window) lets you set field attributes at the table level. Field properties are enforced throughout the database. In some cases, you have the option of setting a particular attribute at either the table or form level. You can set the following attributes at the table level:
  - **Mandatory**: Set the Mandatory attribute for a field that is required for all records.
  - **Display Only**: Set the Display Only attribute for calculated fields or other fields that are not enterable.
  - **Can't Modify**: Set the Can't Modify attribute for fields that accept an initial entry but should not be changed subsequently.
  - **Unique**: Set the Unique attribute for the field that must be used to uniquely identify each record.
  - **Choice list**: You also can associate a choice list with the field. When the choice list is assigned at the table level, it is used on all entry forms and in the Query editor.
- **Relation properties**: The Inspector window contains the Deletion control option that lets you set and enforce referential integrity. You can:
  - Prevent 4D from deleting a related One record if there are related Many records,

- Automatically delete the related Many records if the related One record is deleted,
- Allow the user to delete a related One record even if related Many records exist (turn off referential integrity).

- **Triggers**: You can create triggers that manage the process of loading, saving, and deleting records. Triggers run if a record is loaded, saved, or deleted programmatically, using any form, or during data imports and exports. With triggers, you can enforce complex business rules in a very comprehensive and systematic way. For information on using triggers, see the "Types of 4D Methods" paragraph on page 758 and the section "Triggers" in the *4D Language Reference* manual.
- **Form methods**: You can create form methods that manage the use of the form.

The following sections review the data entry control options that you have at the form level. Use these options in conjunction with data entry controls that are available at the database and table levels.

Setting the Enterable and Mandatory Attributes

The Enterable and Mandatory attributes are similar to the field attributes you set in the Structure editor. If you want these attributes to be different on a particular form, you can modify them in the properties of each field. These attributes can be set in the "Entry" theme of the Property List.

These attributes do not override the field attributes set in the Structure editor. If a field already has the Display Only attribute assigned in the Structure editor, you cannot make it enterable with the Enterable form attribute. If a field already has the Mandatory attribute assigned in the Structure editor, you cannot make it non-mandatory by deselecting the Mandatory form attribute. The Enterable and Mandatory check boxes do not necessarily reflect the attribute settings in the Structure editor.

The Enterable Attribute

Every field is enterable by default. If you want to make a field non-enterable for that form, you can deselect the **Enterable** check box for the object.

A field from a related table may not be enterable if you deselected the **Enterable Related Fields** check box in the Form Wizard. You can make the related field enterable by selecting the **Enterable** check box.

For enterable objects, the **Enterable** check box is checked. You can make an enterable object non-enterable by unchecking the **Enterable** check box. For information about enterable objects, see the section "Enterable and Non-enterable Variables" on page 550.

Note The contents of the Property List are contextual. When the **Enterable** attribute is deselected in the Property List, properties that are related to entry control (Mandatory, Tabable, Entry filter and so on) disappear from the list.

The Mandatory Attribute

No field or enterable object is mandatory by default. To make the field mandatory for all forms, set the Mandatory attribute in the Inspector window in the Structure editor. If you want to make a field or enterable object mandatory for a particular form, you can select the **Mandatory** option in the object properties.

Selecting the **Mandatory** check box makes a field or enterable object mandatory for that form. 4D does not accept a record if the field or object does not contain a value. For information about enterable objects, see the section "Enterable and Non-enterable Variables" on page 550.

The Tabable Attribute

You can set the **Tabable** attribute for each enterable field. When this attribute is selected for a field, it is included in the entry order and will therefore be active when the user presses the **Tab** key.

By default, this attribute is checked for all the fields. If you deselect it, the field will be excluded from the entry order. However, it can still be selected by a mouse click or via a method.

For more information about entry order, refer to the "Data Entry Order" paragraph on page 475.

Hide Focus Rectangle

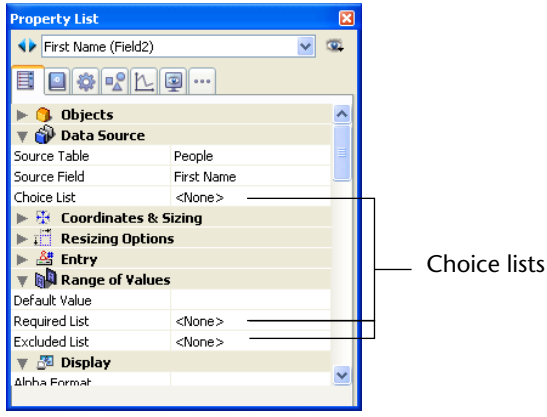
During execution, a field or any enterable object is outlined by a selection rectangle when it has the focus (via the Tab key or a single click). You can hide this rectangle by checking the **Hide focus rectangle** option. Hiding the focus rectangle may be useful in the case of specific interfaces.

Using Choice Lists

You can assign a choice list to a field at either the table or form level. If you want to assign the choice list at the table level, use the Inspector window in the Structure editor. The choice list will then be associated with this field in all the forms and search editors of the database

You can also attach a choice list at the form level. The list can serve as a choice list for that form, as a list of required entries, or as a list of excluded entries.

Choice lists can be associated with a field at the form level in the “Data Source” and “Range of Values” themes of the Property List.



Assigning a list at the form level gives you the freedom to vary data entry constraints from form to form. For example, a field in one input form that is used only by managers can display a comprehensive choice list while the same field in a form that is seen only by drudges has fewer choices.

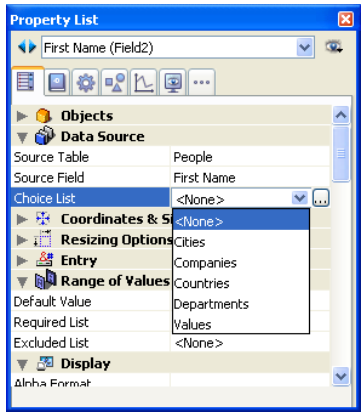
If a field is already assigned a choice list at the table level, you can override the choice list at the form level. If you assign a different list to the field at the form level, then this list is used for that form only.

Before you can assign a choice list, you must have created the list in the Lists editor. For more information about creating choice lists, see the chapter 15 “Lists” on page 903.

► To assign a list to a field or enterable object:

1 **Display the properties of the object in the Property List.**

2 In the "Data Source" and "Range of Values" themes, choose the desired list from the appropriate drop-down list.



4D assigns the selected list(s) to the field or enterable object.

Source Choice Lists

Assigning a source choice list to a field with the Choice List drop-down list ("Data Source" theme) causes 4D to display the choice list automatically during data entry. The choice list appears when the field or enterable object is selected in the form being executed. The user can then select an entry from the list. The entry chosen from the choice list can be overwritten by typing (unless the list is also a required list).

Required Lists

A Required List ("Range of Values" theme) restricts the valid entries to the items on the list. For example, you may want to use a required list for job titles so that valid entries are limited to titles that have been approved by management.

Note Making a list required does not automatically display the list when the field is selected. If you want to display the required list, assign the same list with the Choice List drop-down list ("Data Source" theme).

Excluded Lists

An Excluded List prevents the items on the list from being entered. For example, for a field on an input form used only by drudges, you may want to attach a list of choices that can only be entered by a manager.

Using Entry Filters

An entry filter controls exactly what the user can type during data entry. Unlike the data entry controls discussed earlier in this section, entry filters operate on a character-by-character basis. For example, if a part number always consists of two letters followed by three digits, you can use an entry filter to restrict the user to that pattern. You can even control the particular letters and numbers.

An entry filter operates only during data entry. It has no effect on data display after the user deselects the field. In general, you use entry filters and display formats together. The filter constrains data entry and the format ensures proper display of the value after data entry. For more information, refer to the "Creating Custom Display Formats and Entry Filters" paragraph on page 503.

During data entry, an entry filter evaluates each character as it is typed. If the user attempts to type an invalid character (a number instead of a letter, for example), 4D simply does not accept it. The null character remains unchanged until the user types a valid character.

Entry filters can also be used to display required formatting characters so that the user need not enter them. For example, an American telephone number consists of a three-digit area code, followed by a seven-digit number that is broken up into two groups of three and four digits, respectively. A display format can be used to enclose the area code in parentheses and display a dash after the third digit of the telephone number. When such a format is used, the user does not need to enter the parentheses or the dashes.

An Introduction to Entry Filter Codes

Entry filter codes usually start with an ampersand (&). This character tells 4D to use what follows as an entry filter. If the code starts with a tilde (~), it means the same thing as "&" except that any letter is automatically made uppercase.

The & is usually followed with an "A," an "a," or a "9," meaning allow only uppercase letters (A), allow lowercase and uppercase letters (a), or allow only numbers (9). For example, &9 allows only numbers and &A allows only capital letters.

The number sign (#) tells how many digits or characters are allowed by the code. If the code uses no number signs, the filter allows as many digits or characters as you want. For example, &9 allows as many digits as entered. The filter &9## allows only two digits.

The exclamation point (!) is sometimes used to change which character will appear on screen to indicate the number of characters the user can enter. Without an !, 4D displays an underscore for each digit or character the user can enter. For example, !?&9## displays question marks in both of the places the user will type and it allows only numbers and only two digits.

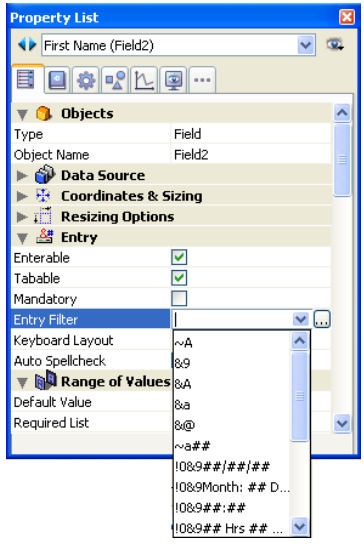
For information about creating entry filters, see the section "Entry Filter Codes" on page 498.

Choosing an Entry Filter

You create the entry filter by choosing a built-in or custom filter from the Entry Filter drop-down list or by typing an entry filter code into the Entry Filter Display area. The Entry Filter drop-down list contains filters for date, time, and alpha fields. The names of any custom filters you create are added to the top of the Entry Filter drop-down list and begin with a vertical bar (|).

For information on creating custom filters, see the section "Creating Custom Display Formats and Entry Filters" on page 503. Most often you will find a suitable entry filter in the drop-down list.

The figure below shows an entry filter being chosen:



Here is a table that explains each of the entry filter choices in the Entry Filter drop-down list.

Entry Filter	Explanation
~A	Allow any letters, but convert to uppercase.
&9	Allow only numbers
&A	Allow only capital letters
&a	Allow only letters (uppercase and lowercase)
&@	Allow only alphanumeric characters. No special characters.
~a##	State name abbreviation (e.g., CA). Allow any two letters, but convert to uppercase.
!0&9##/##/##	Standard date entry format. Display zeros in entry spaces. Allow any numbers.
!0&9 Month: ## Day: ## Year: ##	Custom date entry format. Display zeros in entry spaces. Allow any numbers. Two entries after each word.
!0&9##:##	Time entry format. Limited to hours and minutes. Display zeros in entry spaces. Allow any four numbers, separated by a colon.
!0&9## Hrs ## Mins ## Secs	Time entry format. Display zeros in entry spaces. Allow any two numbers before each word.
!0&9Hrs: ## Mins: ## Secs: ##	Time entry format. Display zeros in entry spaces. Allow any two numbers after each word.
!0&9###-####	Local telephone number format. Display zeros in entry spaces. Allow any number. Three entries, hyphen, four entries.
!_&9(###)!0###-####	Long distance telephone number. Display underscores in first three entry spaces, zeros in remainder.
!0&9###-###-####	Long distance telephone number. Display zeros in entry spaces. Allow any number. Three entries, hyphen, three entries, hyphen, four entries.
!0&9###-##-####	Social Security number. Display zeros in entry spaces. Allow any numbers.

Entry Filter	Explanation
~"A-Z;0-9,;.;-"	Uppercase letters and punctuation. Allow only capital letters, numbers, spaces, commas, periods, and hyphens.
&"a-z;0-9; ;,;.;-"	Upper and lowercase letters and punctuation. Allow lowercase letters, numbers, spaces, commas, periods, and hyphens.
&"0-9;.;-"	Numbers. Allow only numbers, decimal points, and hyphens (minus sign).

You can modify an entry filter after you choose it from the drop-down list. For example, if you want to use a filter that allows upper and lowercase letters, but also need to allow the wildcard character (@), you could choose the filter:

&"a-z;0-9; ;,;.;-"

and change it to:

&"a-z;0-9; ;,;.;-;@"

For more information about modifying entry filters, see the "Creating Custom Display Formats and Entry Filters" paragraph on page 503.

Using Entry Filters and Display Formats Together

You often create a matching display format when you use an entry filter. An entry filter operates only during data entry. It has no effect on how the data is displayed after you tab out of the field.

For example, if you use the Social Security number entry filter (&9###-##-####), you should also choose the matching Social Security number display format (###-##-####). Without the display format, only the numbers, not the hyphens, are displayed in the field. Display formats can be used in both input and output forms and in Quick reports.

Here are some suggested entry filters and matching display formats for common types of fields.

Field Type	Entry Filter	Display Format
State	~a##	(none needed)
Zip Code (standard)	&9#####	(none needed)
Zip Code (extended)	&9#####-####	#####-####

Field Type	Entry Filter	Display Format
Phone number	&9###-#### &9(###) ###-#### &9 ###-###-####	###-#### (###) ###-#### ###-###-####
Social Security number	&9###-##-####	###-##-####
Date	!0&9##/##/## !0&9Month: ## Day: ## Year: ##	(Any Date Format)
Time	!0&9##:## !0&9##Hrs##Mins##Secs !0&9Hrs:##Mins:##Secs:##	(Any Time Format)

You can use display formats on input forms, output forms, and Quick reports. For information about using display formats in Quick reports, refer to the "Assigning the Display Format" paragraph on page 742.

Entry Filter Codes

Often, you create an entry filter simply by choosing it from the Entry Filter drop-down list. If you need a filter for a type of field not covered by the choices in the drop-down list, you can create a filter or modify an existing one.

This section describes how to write the code for an entry filter.

An entry filter code has three parts, in this order:

initiator "argument" placeholders.

The *initiator* informs 4D that the subsequent argument is to be used as a filter during data entry in the field. The *argument* defines the characters to be allowed. The *placeholders* set the places available for the characters.

For example, the following entry filter allows only the letters "a," "b," "c," or "g" to be entered in two places:

&"a;b;c;g"##

In this example, the ampersand (&) is the initiator; the "a;b;c;g" is the argument; and the number signs (#) are the placeholders. The filter can be read as, "Allow the letters 'a', 'b', 'c', or 'g' in two places." Thus the user may enter "ag," "gc," "ba," "ab," "aa," "ac," or any other combination of the four allowed characters.

Entry filters can be combined. The following entry filter allows only the letters “a,” “b,” “c,” or “g” to be entered in two places, followed by the numbers 1, 3, or 8 in one place:

&"a;b;c;g"##&"1;3;8"#

The user must use two of the allowed letters, followed by one of the allowed numbers.

Characters that Initiate a Filter

Two characters initiate a filter: the ampersand (&) and the tilde (~). These characters instruct 4D to use the argument that follows immediately as the filter for the subsequent placeholders.

In addition, the tilde (~) also instructs 4D to make any letters uppercase. It does not prevent a lowercase letter from being typed; it simply changes it to an uppercase letter.

The following entry filters are equivalent in their effects:

&"P"#

~"p"#

The difference between them is that the filter initiated with the ampersand (&) does not accept a lowercase “p.” The filter initiated with the tilde (~) accepts the lowercase “p” but converts it to uppercase.

Because no letters are involved, the following entry filters are equivalent:

&"1;5;8"#

~"1;5;8"#

Arguments

A filter argument follows the initiator and defines the characters that are allowed in the subsequent placeholders. To create a filter argument, surround the characters to be allowed with quotation marks.

Arguments are made up of lowercase letters, uppercase letters, numbers, punctuation marks, and special characters (!@#$%^&*(){}[]":';?><,./`~). If you use a lowercase letter in the argument, only the lowercase form of the letter can be typed by the user. If you use an uppercase letter in the argument, only the uppercase form of the letter can be typed by the user.

An argument may be a single character (a letter or a number), for example, “j,” “J,” or “6.”

An argument may be a set of characters separated by semicolons, for example, “a;r;t” or “1;5.”

An argument may include ranges of characters. A range is defined by the first character, a hyphen, and the last character. Examples are, “a-c” and “1-5.” The “a-c” argument is equivalent to “a;b;c” and “1-5” is equivalent to “1;2;3;4;5.”

An argument may include single letters, single numbers, and one or more ranges, for example, “a;m-z;3;5-9.”

The following table shows useful shorthand versions of arguments. They are used in filters without quotation marks.

Character	Meaning	Equivalent
9	Allow numbers	"0-9"
a	Allow lowercase and uppercase	"a-z;A-Z"
A	Allow uppercase	"A-Z"
@	Allow alphanumeric	"a-z;A-Z;0-9"

The following entry filters are equivalent:

&9#

&"0-9"#

&"1;2;3;4;5;6;7;8;9;0"#

The following entry filters are equivalent:

&a#

&"a-z;A-Z"#

The following entry filters are equivalent:

&A#

&"A-Z"#

Placeholders

The number sign (#) is the only placeholder. You use one number sign for each character the user can enter in the field.

For example, the following entry filter allows the user to enter letters in four places:

&a####

The following entry filter allows the user to enter uppercase letters in three places, followed by numbers in two places:

&A###&9##

If you show no placeholders, the filter code allows any number of characters. The following entry filter allows the user to enter only numbers, but it does not limit the length of the entry:

&9

You can set the maximum number of characters allowed in an Alpha field in the Structure editor.

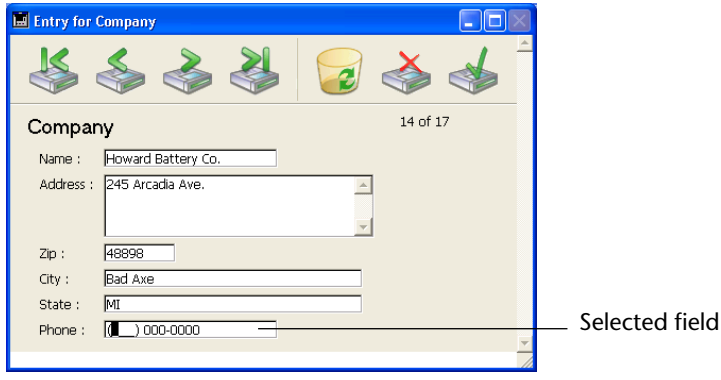
Display Characters

When a field with an entry filter is selected for data entry, 4D displays an underline (_) for each placeholder. As the user types a valid character, each underline is highlighted and replaced with the typed character.

You instruct 4D which character to substitute for the underline by beginning the entry filter with an exclamation point (!) and the character you want.

You can substitute any character for the underline. For example, if you display “XXXX” and the user types only two of the allowed characters (say they are “AA”), the field will contain “AAXX” when the record is saved.

The following illustration shows a selected field displaying underlines and zeros:

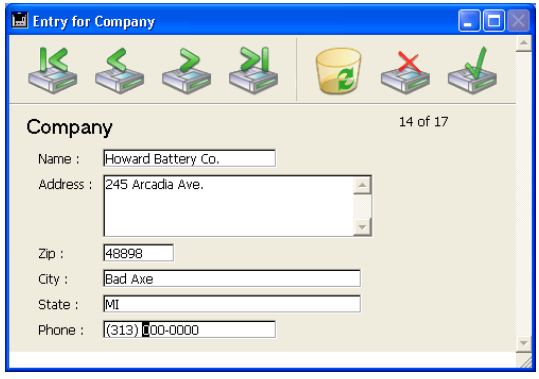


Dead Characters

Any characters, punctuation marks, and spaces can be used as dead characters. Dead characters are displayed during data entry, but they are skipped over by the insertion point and are not entered as part of the data.

The characters you want to use as dead characters are placed before, after, and between placeholders. They are displayed during data entry for clarity.

The phone number entry filter (&9(###) ###-####) uses parentheses, a space, and a dash as dead characters. After you enter a digit immediately preceding a dead character, the insertion point moves directly to the first character following the dead character. The following figure shows how the insertion point skips over the parenthesis and space after the area code to allow the user to enter the next digit in the phone number.



Custom Entry Filter Formats

You can use a custom format to enter an entry filter. All custom formats are automatically displayed in the Entry Filter drop-down list. To use a custom format as an entry filter, choose its name from the Entry Filter drop-down list or type a vertical bar followed by the format name in the Entry Filter Display area. For example, the entry:

|Part Number

installs the custom format named Part Number as the entry filter for the field.

For information about creating custom formats, see the following paragraph.

Creating Custom Display Formats and Entry Filters

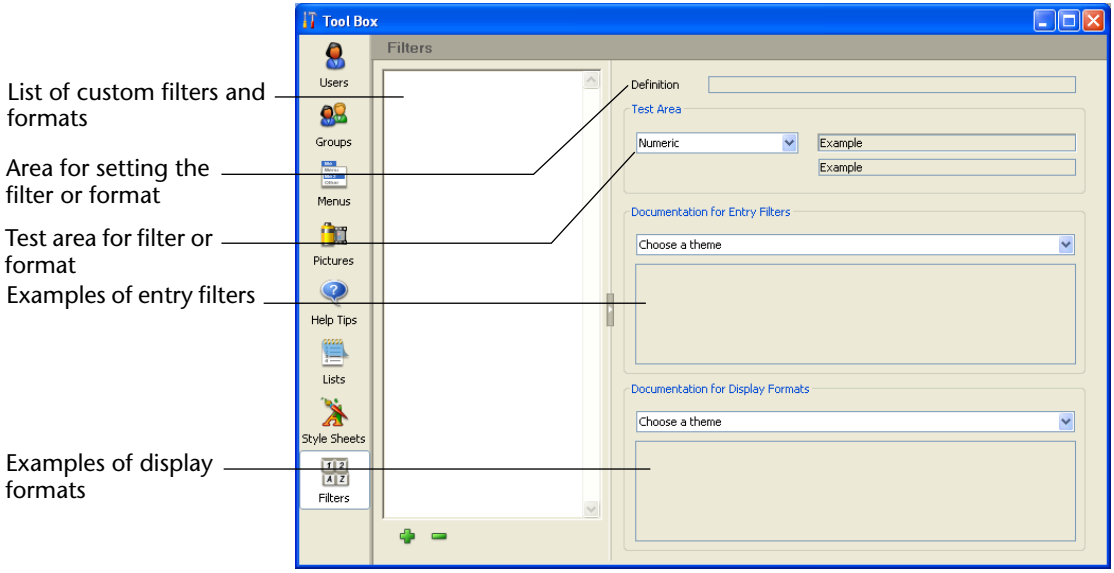
You can create custom display formats and entry filters that you can then refer to by name. You can use a custom format or filter name in place of the code for display formats and entry filters. Custom formats and filters are useful when you use the same display formats or entry filters in several places. If you use fields with the same entry filter in several forms, you can create the entry filter once and specify it by name wherever you need it. In addition, if you decide to change a format or filter, you need only change it in one place and it is updated wherever it is used.

You can also create display formats that correspond to the entry filters and use styles to install them as well.

You can create display formats or entry filters on the **Filters** page of the 4D Tool Box.

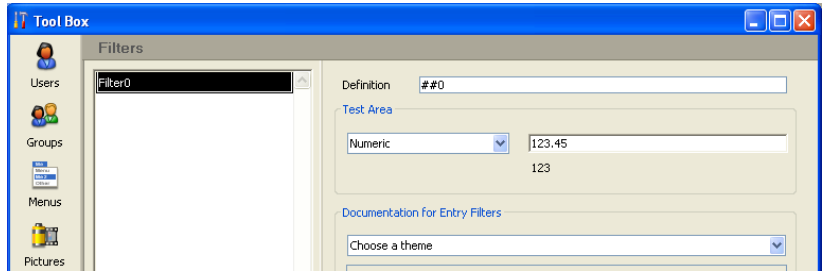
► To create a custom format or filter:

1 **Select Tool Box > Filters and Formats in the Design menu of 4D.**
OR
In the Property List, click on the [...] button to the right of the Entry Filter selection pop-up menu.
The following window appears (empty by default):



2 **Click the add button at the bottom of the list or choose the Add command in the contextual menu of the list (right-click in the list area).**

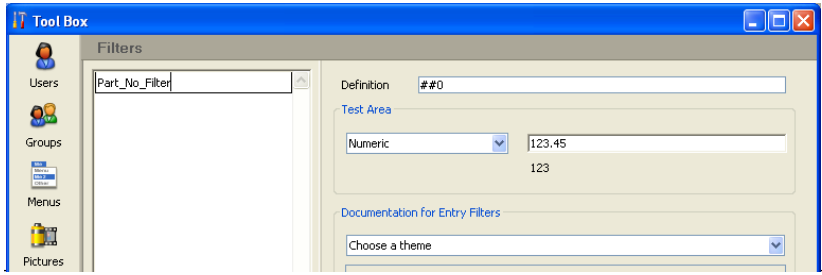
A new item, named "Filter*X*" by default, is added to the list.



3 Type the format or filter name.

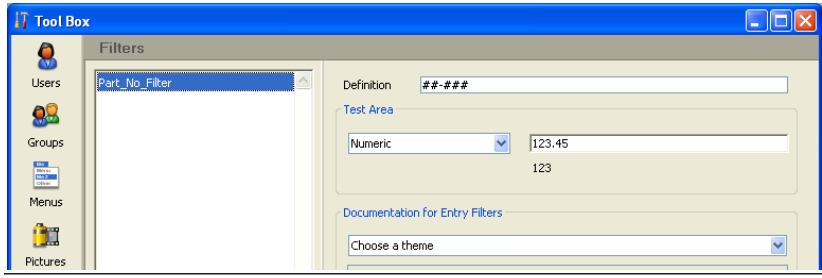
You can edit this name subsequently by pressing the **Alt** (Windows) or **Option** (Mac OS) key and clicking the name of the format or filter you want to change.

You can use up to 255 characters of any type for the name of a format or filter.



Note You can include the word "filter" or "format" in the name to indicate its purpose.

4 Enter the code of the display format or entry filter in the definition area.



Note It is possible to fill in this area by double-clicking in the example areas of the lower part of the window. For more information about the example areas, refer to the "Using the Example Areas" paragraph on page 506.

For example, if you wanted to create a format for a local telephone number, you would use the following:

###-####.

Or, for another example, if you wanted to create a Part Number entry filter for a part number with the format XA-654-1, you would use the following filter:

!X&"A-Z"##-!0&"0-9"###-#

and the corresponding display format is "##-###-#."

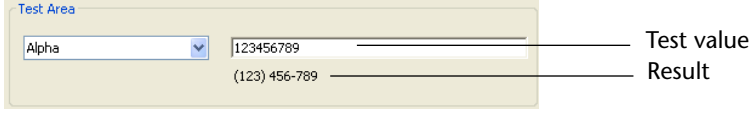
For more information about creating display formats and entry filters, refer to the sections "Data Entry Controls" on page 489 and "Using Entry Filters" on page 494.

5 **(Optional) Test the entry filter or display format in the Test Area.**

This area can be used to check the operation of the filter/format that you are in the process of setting up.
First, you must designate the type of data to which the filter/format will be applied (Alpha, Numeric, Date or Time) using the associated menu.

Once this parameter has been set, enter a test value in the associated area:

- For entry filters, the operation of the filter is checked during data entry,
- For display formats, press the **Enter** or **Carriage return** key after entering the data. The result of applying the display format appears under the entry area:



6 **If you want to create another format or filter, click on the add button again or select the Add command from the contextual menu of the list area (right-click).**
OR
If you want to create a new item based on an existing format or filter, select it and use the Duplicate command of the list area contextual menu or options menu.

You most often create filters and formats in pairs — one for the entry filter and the other for the display format.

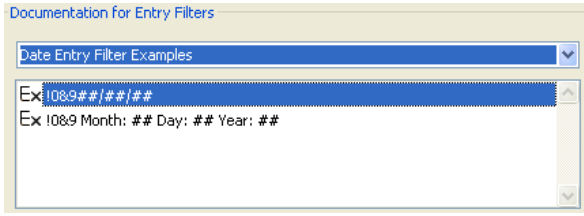
7 When you have finished adding filters and formats, close the dialog box.

You can edit any filter or format by selecting it and changing the name or the code. You can delete any style by selecting it and clicking on the delete button or by selecting the **Delete** command in the contextual menu of the editor.

Using the Example Areas

The "Documentation for Entry Filters" and "Documentation for Display Formats" areas of the filters and formats editing window consist of a menu and a display area.

The menu allows setting a theme for which you wish to obtain information or examples.
When a theme is selected, corresponding characters and information appear in the display area:



You can double-click an example to insert it directly in the Definition area.

Entry filters:

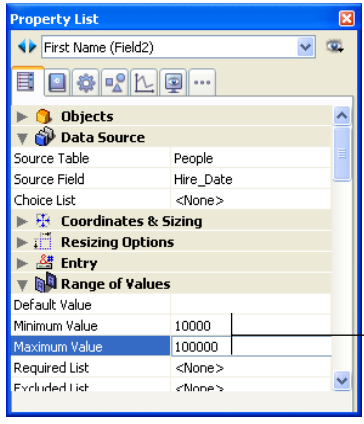
- *Display Characters Initiators*: Modification of placeholders.
- *Initiators*: Filter character initiators.
- *Starting Codes*: Filter start codes.
- *Shorthands*: Filter shortcuts.
- *Placeholders*: Characters used to set possible locations.
- *Main Dead Characters*: Characters ignored in filters.
- *Date Entry Filter Examples*
- *Time Entry Filter Examples*
- *Telephone Entry Filter Examples*
- *Social Security Number Entry Filter Example*
- *Other Entry Filter Examples*.

Display formats:

- *Placeholders*: Characters used to set possible locations.
- *Main Dead Characters*: Characters ignored in filters.
- *Date Display Format Examples*
- *Time Display Format Examples*
- *Telephone Display Format Examples*
- *Social Security Number Display Format Example*.

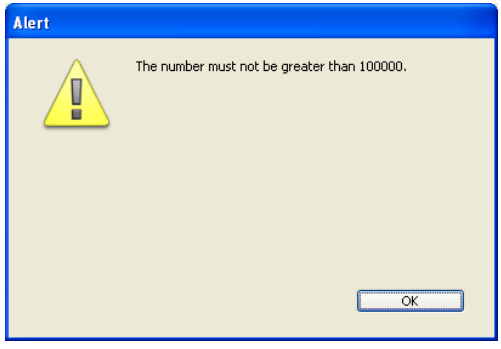
Setting Maximum and Minimum Values

You can restrict a Number, Date, or Time field or enterable object by entering maximum and minimum values in the corresponding entry areas in the “Range of Values” theme of the object properties.



Entry areas for minimum and maximum values

During data entry, if the user enters a value below the minimum or above the maximum, a warning message is displayed.



4D returns the user to the field so that a valid entry can be made.

To set a maximum or minimum value, type the value you want to define the limit. Use the data entry format appropriate for the type of field or enterable object for which you are setting a limit. For example, for a Date field or object, use the date entry format to set the maximum or minimum value.

The values you set are inclusive. That is, if the user enters the same value you have set as a maximum or minimum value, the entry is allowed. Only entries lower than the minimum or higher than the maximum are disallowed. For example, if the value you set as a maximum is 15, the user can enter 15, but not 16.

4D Server Setting a maximum or minimum value changes the maximum or minimum value for all users.

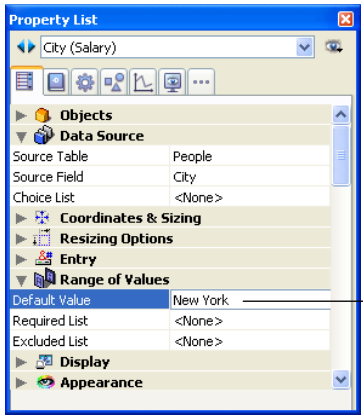
You can also use methods to restrict the values that the user can enter. With a method, you can give more precise and informative feedback to the user, or set minimum or maximum values based on other values in the database. For example, a method can check a customer's credit limit before validating a new transaction.

You can also use a required choice list to create unusual ranges of allowable values. For more information, see the sections "Required Lists" on page 493 and "Creating Lists" on page 908.

Setting Default Values

You can assign a default value to be entered in a field or enterable object. The default value is entered when a new record is first displayed. You can change the value unless the field or entry area has been defined as non-enterable.

You create a default value by typing the value you want in the Default Value entry area in the “Range of Values” theme of the Property List. The default value must be appropriate for the field type.



4D provides stamps for generating default values for the date, time, and sequence number. The date and time are taken from the system date and time. 4D automatically generates any sequence numbers needed. The table below shows the stamp to use to generate default values automatically.

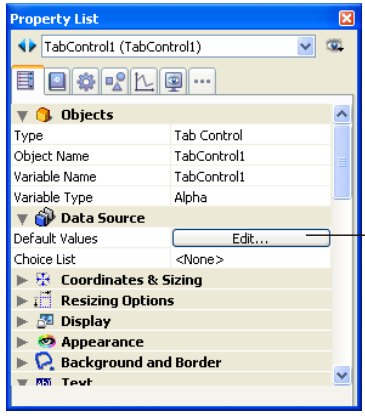
Stamp	Meaning
#D	Current date
#H	Current time
#N	Sequence number

You can use a sequence number to create a unique number for each record. A sequence number is an integer (whole number) that is generated for each new record. The numbers start at one (1) and increase incrementally by one (1). A sequence number is never repeated even if the record it is assigned to is deleted from the table. Each table has its own set of sequence numbers.

Default Lists of Values

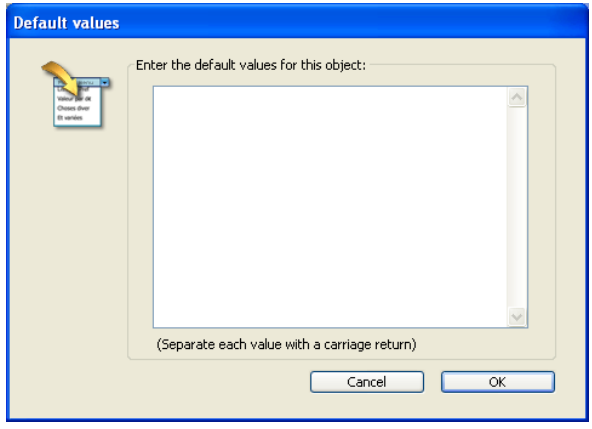
If the object displays a list of values (such as a combo box, scrollable area, pop-up menu, tab control, or drop-down list), you can specify a list of values that will be used as default values. The list will be loaded into the object prior to its being displayed on the form.

For objects that accept a list of default values, the Default Value area becomes a button:



Edit button for Default Values list

When you click this button, the Default values dialog box appears:



Enter the list of default values. Each value should be on a separate line. Click **OK** to put away the Default values dialog box and return to the Property List window.

When you enter default values into the Default values dialog box, they are automatically loaded into an array whose name is the name of the object. Using the language, you can manage the object by referring to this array.

4D Server Setting a default value in the Property List window sets the default value for all users.

Using a List to Set Default Values

If the object is a hierarchical list or a tab control, you can use a list that you created using the Lists editor to set default values.

► To set default values using a list:

1 **Expand the "Data Source" theme of the Property List.**

2 **Select the desired list from the Choice List drop-down list.**

Setting Default Values Using the Language

You can also set default values using a method. For objects that accept one, you can assign the default value when the On Load event executes in the object or form method.

For objects that accept lists, you can enter the default values using the Lists editor and then use the Load list command in order to create a hierarchical list. You can then work with the list and its contents using the commands of the "Hierarchical Lists" theme. You can load the lists into a hierarchical list when the On Load form event occurs or load all lists in the On Startup database method.

Keyboard Layout

You can assign a specific keyboard layout property to enterable fields and variables in your forms.

Compatibility Note: This option is only taken into account when the database is running in non-Unicode mode (see the "Design Compatibility" paragraph on page 173). Starting with version 11 of 4D, it is recommended to work in Unicode mode, especially in the case of applications working with non-Roman characters. In Unicode mode, this property is ignored.

This allows data entry and display using a different language script than that of the system — mainly useful for languages which might use several alphabets, such as Japanese. It "forces" the script to be used for internal conversion of the entered characters. It also causes the keyboard to change automatically when the object receives the cursor. If the user changes the current keyboard during the 4D session without this property being set, entry in forms will be possible but the characters will not be stored correctly (they will disappear when the cursor leaves the area).

To associate a specific keyboard layout with a field or variable, choose a value from the **Keyboard Layout** menu in the Property List ("Entry" theme).

When the default value (None) is used, 4D keeps the current keyboard layout.

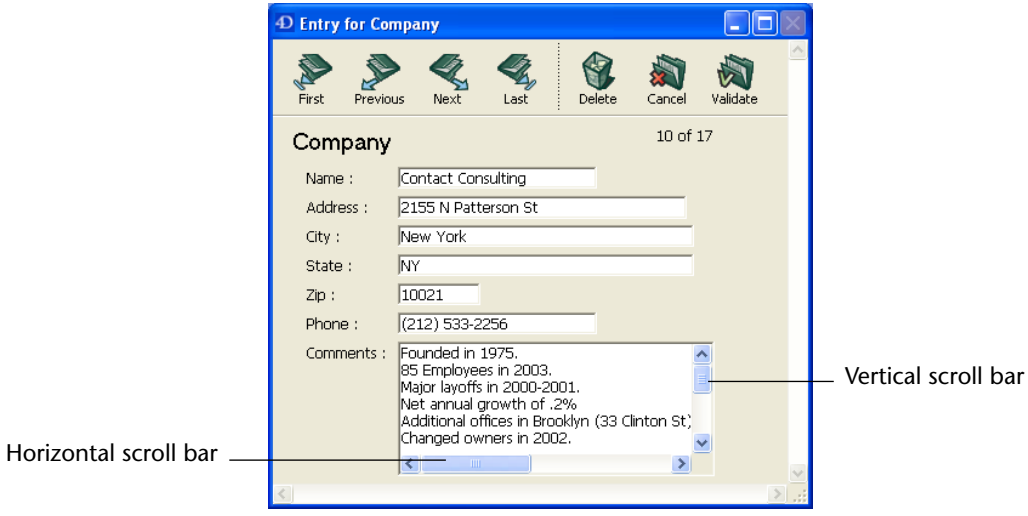
Note This property only affects data entry and display. Data processing (sorts, queries, etc.) is always carried out using the current script of the system. 4D automatically uses the script that corresponds to the current system language for data entry, display and processing (such as sorting).

Adding Scroll Bars to Text or Picture Objects

Two types of objects can have scroll bars associated with them: fields and variables of the Text type and fields and variables of the Picture type. These properties can be managed using the Property List or via programming using the SET SCROLLBAR VISIBLE command.

Text Type Objects

Text fields and variables can contain up to 2 GB of text. 4D allows you to attach scroll bars so that the user can display all the information. The figure below shows an input form with a text area that has scroll bars:



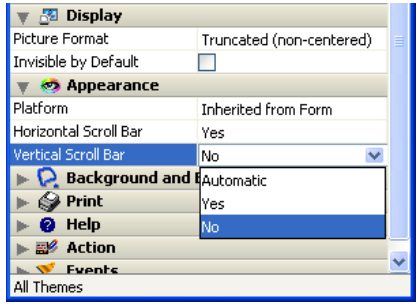
Note If a text field or enterable object does not have a scroll bar, the user can scroll the information using the arrow keys.

The **Horizontal Scroll Bar** and **Vertical Scroll Bar** properties are active in the "Appearance" theme of the Property List for Text type objects.

Picture Type Objects

Picture type objects can have scroll bars when the display format of the picture is set to "**Truncated (non-centered)**." For more information about picture display formats, please refer to the "Picture Field Formats" paragraph on page 534.

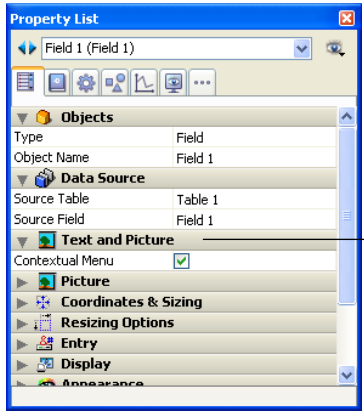
In this case, the **Horizontal Scroll Bar** and **Vertical Scroll Bar** properties are active in the Project List for the picture object. Each property is set using a menu with three options:



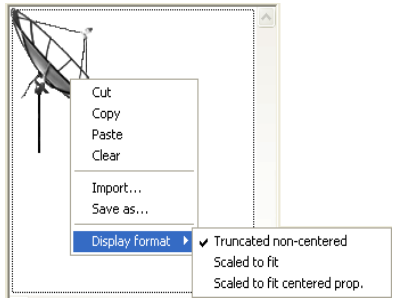
- **Yes**: The scrollbar is always visible, even when it is not necessary (in other words, when the size of the picture is smaller than the frame).
- **No**: The scrollbar is never visible.
- **Automatic**: The scrollbar appears automatically whenever necessary (in other words, when the picture size is greater than the frame).

Associating a Contextual Menu with Pictures

It is possible to associate an automatic contextual menu with Picture type fields and variables. To do this, simply check the **Contextual Menu** option in the "Text and Picture" theme of the Property List:



Once this option has been checked, the area will have a contextual menu when the form is executed in the Design or Application environment. The user can access editing and display commands by clicking on the picture with the **right mouse button**:



In addition to standard editing commands (**Cut**, **Copy**, **Paste** and **Clear**), the menu also contains the **Import...** command, which can be used to import a picture stored in a file, as well as the **Save as...** command, which can be used to save the picture to disk. These two commands take advantage of native picture management: they can be used respectively to open and save pictures in any native format supported by 4D.

The menu can also be used to modify the display format of the picture: the **Truncated non-centered**, **Scaled to fit** and **Scaled to fit centered prop.** options are provided. The modification of the display format using this menu is temporary; it is not saved with the record. For more information about picture display formats, please refer to the "Picture Field Formats" paragraph on page 534.

Note If the picture field or variable is not enterable, only the **Copy**, **Save as...** and the formatting commands are available.

URL Detection and Activation

When a field or variable of the Alpha or Text type contains a URL (string starting with http, ftp, www or mailto), it is automatically detected; under Windows, it is displayed in blue and underlined:

The URL can then be executed directly in the default Web browser by a **Ctrl+Click** (Windows) or **Command+Click** (Mac OS) in the area.

Note E-mail addresses must be written in the mailto:address form.

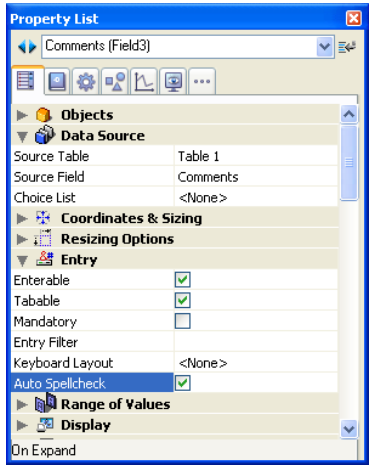
Using the Spell-check

4D includes an integrated spell-check utility that is available in several languages. Alpha and Text type variables and fields can be checked, as well as 4D Write documents.

Activating the Spell-check

There are two ways to activate the spell-check for a field or variable of the Alpha or Text type:

- Check the **Auto Spellcheck** property ("Entry" theme) for the field or variable:

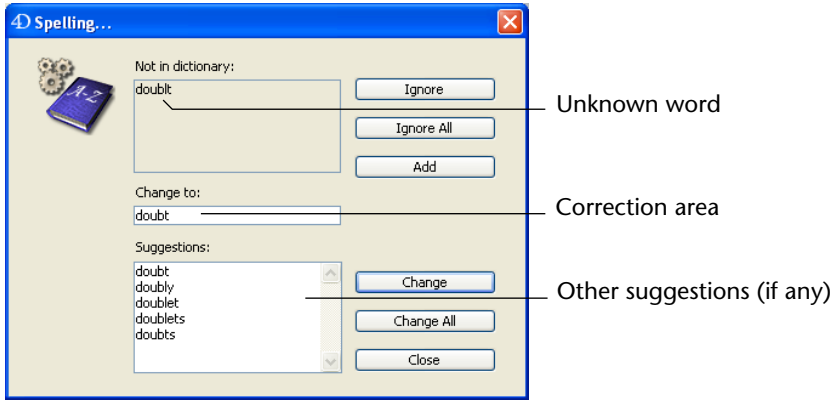


In this case, the spell-check is done automatically during entry, once the object loses the focus.

- Execute the SPELL CHECKING command for each object to be checked. For more information on this command, refer to the *4D Language Reference* manual.

Detecting an Unknown Word

Regardless of how the spell-check feature is activated, the following dialog box appears if an unknown word is detected:



Several buttons are available:

- **Ignore**: The unknown word is kept as is.
- **Ignore all**: The unknown word is kept as is and all other occurrences of the word in the area are also kept.
- **Add**: The unknown word is added to the dictionary; it will no longer be detected by the spell-check feature.
- **Change**: The unknown word is replaced by the word shown in the correction area.
- **Change all**: The unknown word and all other occurrences of the word are replaced with the word shown in the correction area.
- **Close**: No correction is made and the dialog box is closed.

Note When you click a button other than the **Close** button, the spell-check continues until it reaches the end of the area.

Managing Dictionaries

The 4D spell-check can use four dictionaries: English, French, German, and Spanish. By default, 4D uses the dictionary corresponding to the current language of the application.

You can, however, force the opening of a dictionary in a different language using the SET DICTIONARY command. For more information about this command, refer to the *4D Language Reference* manual.

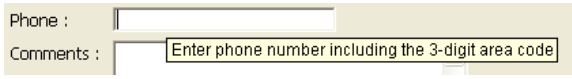
4D lets you set up and use **specialized dictionaries** for each language, including lists of custom words accepted by the spell-check. For more information about setting up these dictionaries, please refer to the Appendix D on page 1303.

Adding Help Tips to a Field or Object

You can add a help message to fields and active objects in your forms to help users work with your database more productively. Help messages will appear as tips.

For example, you can create a help message for a Date field which reminds the user to include a separator such as the slash mark (/) between the month, day, and year when entering data.

The help tip appears in the form whenever the mouse moves over the field or object.



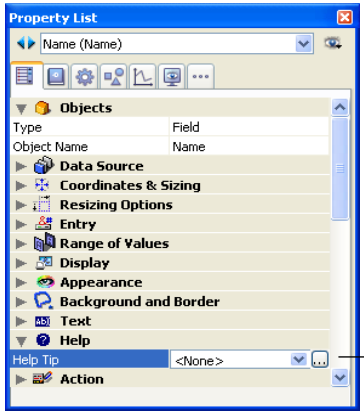
Note You can also associate a help message with a field at the level of the database structure. In this case, the help tip of the field is displayed in every form where it appears. For more information, refer to "Help Tips" on page 285.
If two different help tips are associated with the same field in these two locations, priority is granted to the help tip set at the form level.

Associating a Help Tip with an Object

You can associate a help tip with any active object in your form.

► To associate a help tip with an object:

1 Select the object and expand the "Help" theme of the Property List.



2 Choose a message from the Help Tip menu.

The help tip must have been previously specified in the Help Tip editor of 4D.

OR

Enter the help message directly in the Help Tip menu.

This allows you to take advantage of XLIFF architecture. You can enter an XLIFF reference here in order to display a message in the application language (for more information about XLIFF, please refer to Appendix C on page 1293).

The message you select or enter will appear as a help tip for the field or object selected in the form.

To delete the association of a help tip with the selected object, choose **None** in the Property List.

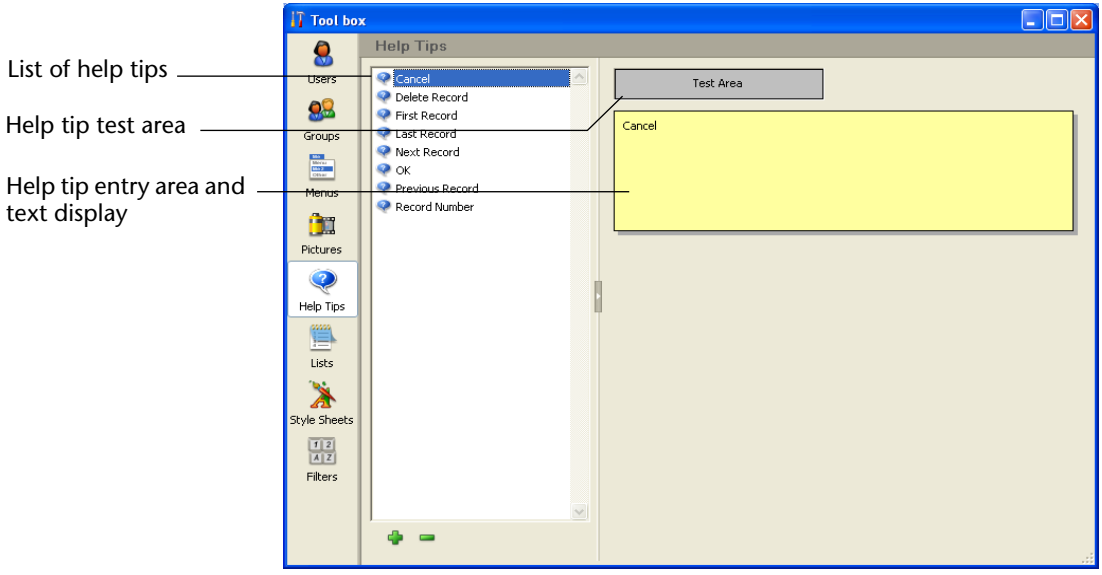
Creating Help Tips

You can create and test help tips on the **Help Tips** page of the 4D tool box.

▶ To create a help tip to be used in forms:

1 **Select Tool Box > Help Tips in the Design menu of 4D.**
OR
In the Property List, click on the [...] button to the right of the help tip selection pop-up menu.

The Help Tips page appears. It contains the help tips automatically generated by the Form Wizard or for the default forms:



2 **Click on the add button at the bottom of the list or choose Add in the contextual menu of the list area (right-click in the list area).**

A new item, named "Help text #1" by default, is added to the list.

3 **(Optional) Enter a name for the help tip.**

The name of the help tip is used for reference in the application; it does not appear to the user. You can use up to 15 characters.

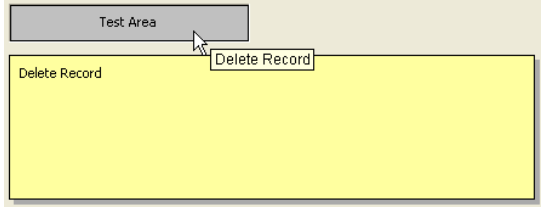
You can modify this name subsequently by holding down the **Alt** (under Windows) or **Option** (under Mac OS) key before clicking on the name of the help tip so that it becomes editable. You can also use the **Rename** command of the contextual menu or options menu in the list area.
If you do not modify the default name, 4D will create other help tips named *Help text#2*, *Help text#3,* and so on by default.

Note Help tips that are entered directly in the Property List are not added to the list of help tips.

The help texts are listed in alphabetical order. When you add a new help tip, the contents of the list is automatically updated accordingly.

4 **Press the Tab key or click in the entry area and type the text of the help tip.**
You can enter up to 255 characters and use dynamic references but not XLIFF references (see the following paragraph).

5 **(Optional) Test the display of the help tip by placing the mouse cursor over the test area, without clicking.**



This way you can view the help tip as it will appear in the database.

6 **If you want to create another help tip, click on the add button again or select the Add command in the contextual menu of the list area (right-click).**
OR
If you want to create a new help tip based on an existing one, select it in the list and use the Duplicate command of the contextual menu or the options menu in the list area.

Using Dynamic References

You can insert dynamic references for variables, fields and titles into the help tips specified using the help tips editor or in help tips entered directly in the Property List. The following dynamic elements can be used:

- An **STR# resource reference**: The syntax to apply is ":16000,2" where 16000 is the resource number and 2 is its element.

- An **XLIFF reference** (only for help tips entered using the Property List): several syntaxes are accepted. For more information, please refer to the Appendix C on page 1293.
- A **table or field label**: The syntax to apply is <?[TableNum]FieldNum> or <?[TableName]FieldName>. For more information, refer to "Inserting Dynamic Table and Field Names" on page 487.
- A **variable or a field**: The syntax to apply is <VariableName> or <[TableName]FieldName>. The current value of the field or variable will be displayed in the help message.
  You can, for example, enter the following text in a help message: "Enter <[Family]First_Name>'s age in this area." During use, 4D will replace the field reference by the current value for the [Family]First_Name field.
  Dynamic references of the variable or field type cannot be displayed in the test area of the Help Tips editor.

Modifying or Deleting a Help Tip

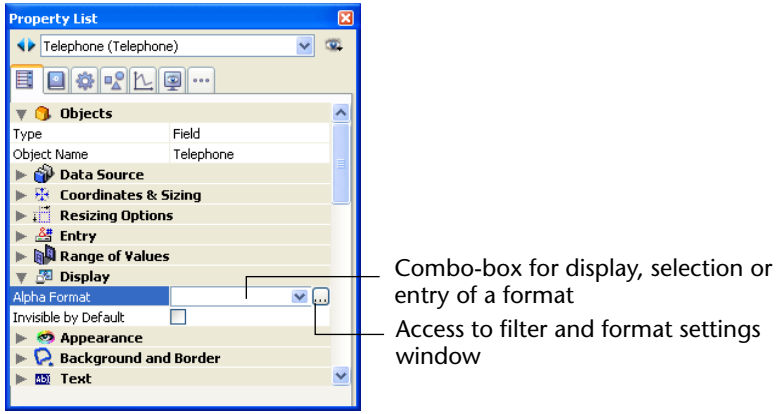
You can modify any help tip by selecting it and changing its contents. Press the **Tab** key or click outside of the area to validate your changes. You can delete any help tip by selecting it and clicking on the delete button or by choosing the **Delete** command in the contextual menu of the editor.

Display Formats

The display formats provided by 4D give you many choices for screen display and printing. Display formats can be applied to both fields and enterable or non-enterable areas (variables). The format you use to display the contents of a field does not affect the actual value stored by 4D.

The display format for a field can be different in each form. For example, you may want to show a value without dollar signs in an input form and display it with dollar signs in an output form.

You set display formats in the “Display” theme of the Property List.



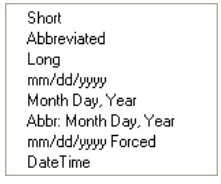
Different formats appear in the selection combo-box depending on the type of field or variable you select.

Note You set the type of a variable using the **Variable Type** property in the “Objects” theme of the Property List.

The built-in formats for the type set are always displayed. Any display formats that were added using the Formats and Filters editor of the tool box appear at the start of the list, preceded by a vertical bar | (see the “Creating Custom Display Formats and Entry Filters” paragraph on page 503).

Date Field Formats

Date formats control the way dates appear when displayed or printed. For data entry, you enter dates in the *MM/DD/YYYY* format, regardless of the display format you have chosen.
The figure below shows date formats in the Format pop-up menu.



Note Unlike Alpha and Number formats, display formats for dates must only be selected among the 4D built-in formats.

The table below shows the Date field display formats and gives an example of each format.

Choice	Example
System date short	3/25/99
System date abbreviated[1]	Wed, Mar 25, 1999
System date long	Wednesday, March 25, 1999
Internal date short special	03/25/99 but 04/25/2032[2]
Internal date long	March 25, 1999
Internal date abbreviated[1]	Mar 25, 1999
Internal date short	03/25/1999
ISO Date Time[3]	1999-03-25T00:00:00

1. To avoid ambiguity and in accordance with current practice, the abbreviated date formats now display “jun” for June “jul” for July (instead of “jui” for both as previously). This particularity only applies to French versions of 4D.
2. The year is displayed using two digits when it belongs to the interval (1930;2029) otherwise it will be displayed using four digits. This is by default but it can be modified using the SET DEFAULT CENTURY command.
3. The ISO Date Time format corresponds to the XML date and time representation standard (ISO8601). It is mainly intended to be used when importing/exporting data in XML format and in Web Services.

Note Regardless of the display format, if the year is entered with two digits then 4D assumes the century to be the 21st if the year belongs to the interval (00;29) and the 20th if it belongs to the interval (30;99). This is the default setting but it can be modified using the SET DEFAULT CENTURY command.

Blank if Null

By default, a null date is generally displayed as “00/00/00.” The new **Blank if null** option of the Property List (“Display” theme) can be used to display an empty area if the date is null or contains the SQL NULL attribute.

Time Field Formats

Time formats control the way times appear when displayed or printed. For data entry, you enter times in the 24-hour *HH:MM:SS* format or the 12-hour *HH:MM:SS AM/PM* format, regardless of the display format you have chosen.

Note Unlike Alpha and Number display formats, the Time display format must only be selected in the Format pop-up menu.

The table below shows the Time field display formats and gives examples:

Format	Comments	Example for 04:30:25
HH:MM:SS		04:30:25
HH:MM		04:30
Hour Min Sec		4 hours 30 minutes 25 seconds
Hour Min		4 hours 30 minutes
HH:MM AM/PM		4:30 a.m.
MM SS	Time expressed as a duration from 00:00:00	270:25
Min Sec		270 Minutes 25 Seconds
ISO Date Time	Corresponds to the XML standard for representing time-related data. It is mainly intended to be used when importing/exporting data in XML format.	0000-00-00T04:30:25
System time short	Standard time format defined in the system	04:30:25
System time long abbreviated	*Mac OS only*: Long time format defined in the system Under Windows, this format is the same as the System time short format.	4•30•25 AM
System time long		4:30:25 AM HNEC

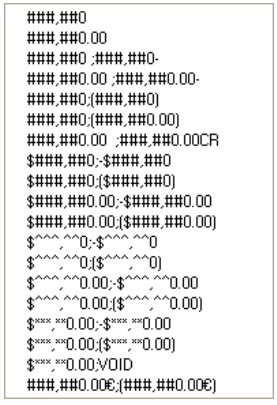
Blank if Null

By default, a null time is displayed for example as “00:00:00” (the display depends on the format applied to the object). The **Blank if null** option of the Property List (“Display” theme) can be used to display an empty area if the time is null or contains the SQL NULL attribute.

Number Field Formats

Preliminary Note: number fields include the Integer, Long integer, Integer 64 bits, Real and Float types.
Number formats control the way numbers appear when displayed or printed. For data entry, you enter only the numbers (including a decimal point or minus sign if necessary), regardless of the display format you have chosen.

The following illustration shows the number formats in the Format pop-up menu:



Note This list can be different in converted databases, see the "System Preferences for Separators" paragraph on page 526.

You can choose the format from the pop-up menu or type it and/or modify it in the combo box of the Property List. You can also select a custom format name set in the Filters and Formats editor of the tool box. In this case, the format cannot be modified in the object properties. You can access the editor by clicking on the [...] button to the right of the format combo box.
A number field can use any format including a custom format. Creating a custom number display format is discussed in the following section.

Creating a Custom Number Format

In each of the number display formats, the number sign (#), zero (0), caret (^), and asterisk (*) are used as placeholders. You create your own number formats by using one placeholder for each digit you expect to display.

For example, if you want to display three-digit numbers, you could use the format ###. If the user enters more digits than the format allows, 4D displays <<< in the field to indicate that more digits were entered than the number of digits specified in the display format. If the user enters a negative number, the leftmost character is displayed as a minus sign (unless a negative display format has been specified). If ##0 is the format, minus 26 is displayed as –26 and minus 260 is displayed as <<< because the minus sign occupies a placeholder and there are only three placeholders.

Note No matter what the display format, 4D accepts and stores the number entered in the field. No information is lost.

Each placeholder character has a different effect on the display of leading or trailing zeros. A *leading zero* is a zero that starts a number before the decimal point; a *trailing zero* is a zero that ends a number after the decimal point.

Suppose you use the format ##0 to display three digits. If the user enters nothing in the field, the field displays 0. If the user enters 26, the field displays 26.

The table below explains the effect of each placeholder on leading and trailing zeros.

Placeholder	Effect for leading or trailing zero
#	Displays nothing
0	Displays 0
^	Displays a space[1]
*	Displays an asterisk

1. The caret (^) generates a space character that occupies the same width as a digit in most fonts.

System Preferences for Separators

The numeric display formats (except for scientific notations) automatically based on regional system parameters. 4D replaces the "." and "," characters by, respectively, the decimal separator and the thousand separator defined in the operating system. The period and comma are thus considered as *placeholder characters*, following the example of 0 or #.

Compatibility Note: In previous versions of 4D, numeric display formats do not take the regional parameters of the system into account. To use regional parameters in databases converted from versions prior to v11, you must check the "**Use system settings in numeric formats**" option (see the "Design Compatibility" paragraph on page 173). The following table compares the results depending on the state of the compatibility option:

Value to be displayed	15353.33	
Display format	###,##0.00	
"Use system settings in numeric formats" option in converted databases	**Not checked (default)**	**Checked**
Display on a US system *thousands separator = ,* *decimal separator= .*	15,353.33	15,353.33
Display on a French system *thousands separator = <space>* *decimal separator = ,*	<<<<<<<< (error)	15 353,33
Display on a Swiss system *thousands separator = '* *decimal separator = ,*	<<<<<<<< (error)	15'353,33

Starting with 4D v11, databases that are created have the same functioning as that when this option is **Checked**.

Decimal Points and Other Display Characters

You can use one decimal point in the format. If you want the decimal to display regardless of whether the user types it in, it must be placed between zeros.

You can use any other characters in the format. When used alone, or placed before or after placeholders, the characters always appear. For example, if you use the following format:

$##0

a dollar sign always appears because it is placed before the placeholders.

If characters are placed between placeholders, they appear only if digits are displayed on both sides. For example, if you define the format:

###,##0

a comma appears only if the user enters at least four digits.

Spaces are treated as characters in number display formats.

Formats for Positive, Negative, and Zero

A number display format can have up to three parts allowing you to specify display formats for positive, negative, and zero values. You specify the three parts by separating them with semicolons as shown below:

Positive;Negative;Zero

You do not have to specify all three parts of the format. If you use just one part, 4D uses it for all numbers, placing a minus sign in front of negative numbers. If you use two parts, 4D uses the first part for positive numbers and zero and the second part for negative numbers. If you use three parts, the first is for positive numbers, the second for negative numbers, and the third for zero.

Note The third part (zero) is not interpreted and does not accept replacement characters. If you enter ###;###;# the zero value will be displayed "#". In other words, what you actually enter is what will be displayed for the zero value.

Here is an example of a number display format that shows dollar signs and commas, places negative values in parentheses, and does not display zeros:

$###,##0.00;($###,##0.00);

Notice that the presence of the second semicolon instructs 4D to use nothing to display zero.

The following format is similar except that the absence of the second semicolon instructs 4D to use the positive number format for zero:

$###,##0.00;($###,##0.00)

In this case, the display for zero would be $0.00.

Scientific Notation

If you want to display numbers in scientific notation, use the ampersand (&) followed by a number to specify the number of digits you want to display. For example, the format

&3

would display 759.62 as

7.60e+2.

The scientific notation format is the only format that will automatically round the displayed number. Note in the example above that the number is rounded up to 7.60e+2 instead of truncating to 7.59e+2.

Hexadecimal Formats

You can display a number in hexadecimal using the following display formats:

- **&x**: This format displays hexadecimal numbers using the "0xFFFF" format.
- **&$**: This format displays hexadecimal numbers using the "$FFFF" format.

Displaying a Number as a Time

You can display a number as a time (with a time format) by using "&/" followed by a digit. Time is determined by calculating the number of seconds since midnight that the value represents. The digit in the format corresponds to the order in which the time format appears in the Format drop-down menu.

For example, the format:

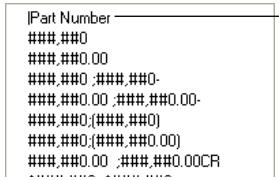
&/5

corresponds to the 5^{th} time format in the pop-up menu, specifically the AM/PM time. A number field with this format would display 25000 as:

6:56 AM.

Custom Number Formats

You can use custom number display formats, created in the Filters and Formats editor of the tool box. The custom format (as well as filter) names appear at the beginning of the alpha and number format lists, preceded by a vertical bar (|):



For more information about creating custom formats, refer to "Creating Custom Display Formats and Entry Filters" on page 503.

Examples

The following table shows how different formats affect the display of numbers. The three columns — Positive, Negative, and Zero — each show how 1,234.50, –1,234.50, and 0 would be displayed.

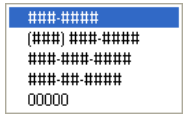
Format Entered	Positive	Negative	Zero
###	<<<	<<<	
####	1234	<<<<	
#######	1234	–1234	
#####.##	1234.5	–1234.5	
####0.00	1234.50	–1234.50	0.00
#####0	1234	–1234	0
+#####0;–#####0;0	+1234	–1234	0
#####0DB;#####0CR; 0	1234DB	1234CR	0
#####0 ;(#####0)	1234	(1234)	0
###,##0	1,234	–1,234	0
##,##0.00	1,234.50	–1,234.50	0.00
^^^^^^^	1234	–1234	
^^^^^^0	1234	–1234	0
^^^,^^0	1,234	–1,234	0
^^,^^0.00	1,234.50	–1,234.50	0.00
*******	***1234	**–1234	*******
******0	***1234	**–1234	******0
***,**0	**1,234	*–1,234	******0
,0.00	*1,234.50	–1,234.50	*****0.00

Format Entered	Positive	Negative	Zero
$*,**0.00;–$*,**0.00	$1,234.50	–$1,234.50	$****0.00
$^^^^0	$ 1234	$–1234	$ 0
$^^^0;–$^^^0	$1234	–$1234	$ 0
$^^^0 ;($^^^0)	$1234	($1234)	$ 0
$^,^^0.00 ;($^,^^0.00)	$1,234.50	($1,234.50)	$ 0.00
&2	1.2e+3	-1.2e+3	0.0e+0
&5	1.23450e+3	-1.23450e+3	0.00000

Alpha Field Formats

Alpha formats control the way the alphanumeric fields and variables appear when displayed or printed.

The figure below shows an alpha format being chosen from the format pop-up menu.



You can choose a format from this list or type it and/or modify it in the combo box of the Property List. The Format pop-up menu contains formats for some of the most common alpha fields that require formats: US telephone numbers (local and long distance), Social Security numbers, and zip codes.

The following table shows the Alpha field formats and the types of fields they are typically used for.

Format	Field type
###-####	Telephone number (local)
(###) ###-####	Telephone number (long distance)
###-###-####	Telephone number (long distance)
###-##-####	Social Security number
00000	Zip Code

You can also select a custom format name set in the Filters and Formats editor of the tool box. In this case, the format cannot be modified in the object properties. You can access the editor by clicking on the [...] button to the right of the formats combo box.

The number sign (#) is the placeholder for an Alpha field display format. You can include the appropriate dashes, hyphens, spaces, and any other punctuation marks that you want to display. You use the actual punctuation marks you want and the number sign for each character you want to display.

For example, consider a part number with a format such as:

RB-1762-1.

The alpha format would be:

##-####-#.

When the user enters "RB17621," the field displays:

RB-1762-1.

The field actually contains "RB17621."

If the user enters more characters than the format allows, 4D displays the last characters. For example, if the format is:

(###)

and the user enters "HAPPY," the field displays:

(PPY).

The field actually contains "HAPPY." 4D accepts and stores the entire entry no matter what the display format. No information is lost.

Custom Alpha Formats

You can use a custom format for Alpha fields. Any custom formats or filters that you have created using the Filters and Formats editor of the tool box are automatically added to the beginning of the alpha and number format lists, preceded by a vertical bar (|). You can choose a custom format just as you would choose a built-in format.

For information about creating custom formats, see the section "Creating Custom Display Formats and Entry Filters" on page 503.

Boolean Field Formats

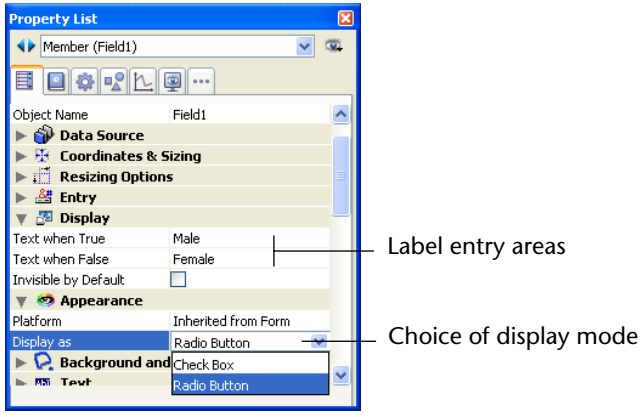
Boolean fields can contain one of two values: TRUE or FALSE. A Boolean field can be displayed as either a pair of radio buttons or as a check box.

If you do not create a display format for a Boolean field, 4D automatically displays the field as a pair of radio buttons labeled Yes and No.

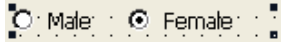
You can specify the form and label(s) of a Boolean field in the field properties. If you want to display only the buttons or check boxes and not the field name, you can delete the field label in the form.

Formatting a Boolean Field as Radio Buttons

The appearance of Boolean fields is set using the "Display as" drop-down list in the "Appearance" theme. Once you have selected **Radio Button** from that list, you can enter the labels for each value in the **Text when True** and **Text when False** entry areas under the "Display" theme.



The buttons are displayed in the Form editor side by side as shown below.

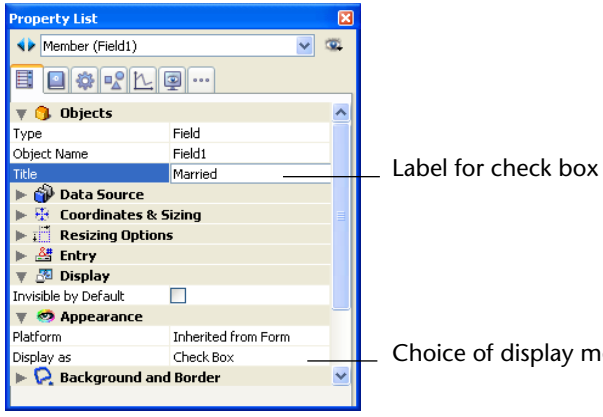


If you use labels with different first letters, you can select the radio button by typing the first letter during data entry. For example, you can press "M" to select Male or "F" to select Female when the field is selected.

The following rules apply when the field is being used for data storage: if the first button is selected, the field is true; if the second button is selected, the field is false. The field is false by default.

Formatting a Boolean Field as a Check Box

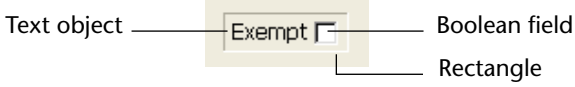
Choose **Check box** from the "Display as" list located in the "Appearance" theme. Once you have selected that option, an entry area labeled "Title" is displayed in the "Objects" theme. This is the entry area in which you enter the label of the check box. The default label is the field's name.



The following rules apply when the field is being used for data storage: if the check box is selected, the field is True; if the check box is deselected, the field is False. The field is False by default.

During execution, this field is displayed as a check box:

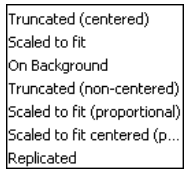
You can format a Boolean field as a check box with no label by entering a space in the Title area and setting a transparent border line style. In this case, you add the label for the check box as a separate object. You can then place the label wherever you want, draw a rectangle and insert dynamic references in the label (refer to "Working with Static Text Areas" on page 445). In the following example, the Boolean field is on top of the text object.



Picture Field Formats

Picture field formats control how pictures appear when displayed or printed. For data entry, the user always enters pictures by pasting them from the Clipboard or by drag and drop, regardless of the display format.

The figure below shows the picture formats available in the Picture Format list of the Property List (“Display” theme).



The truncation and scaling options do not affect the picture itself. The contents of a Picture field are always saved. Only the display on the particular form is affected by the picture display format.

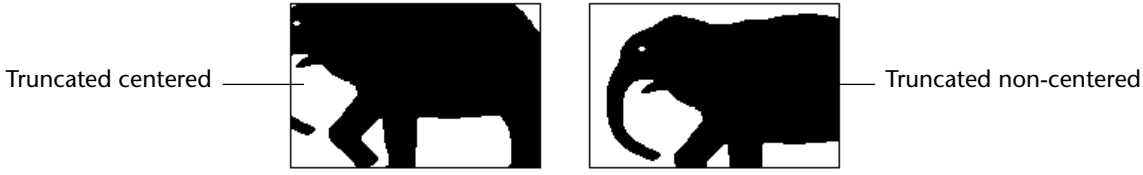
Truncated (Centered and Non-centered)

The Truncated (centered) format causes 4D to center the picture in the field and crop any portion that does not fit within the field area. 4D crops equally from each edge and from the top and bottom.

The Truncated (non-centered) format causes 4D to place the upper-left corner of the picture in the upper-left corner of the field and crop any portion that does not fit within the field area. 4D crops from the right and bottom.

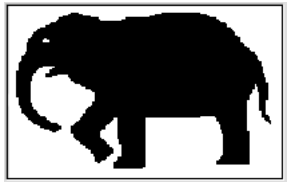
Note When the picture format is Truncated (non-centered), it is possible to add scroll bars to the field or variable area. For more information, please refer to the “Adding Scroll Bars to Text or Picture Objects” paragraph on page 512.

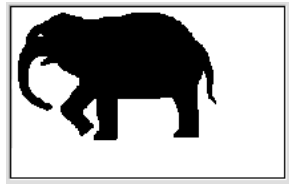
The figure below compares the Truncated centered and non-centered formats.

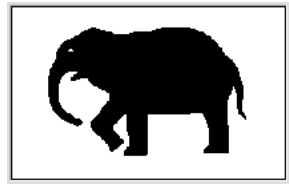


Scaled to Fit

The Scaled to fit formats cause 4D to resize the picture to fit the dimensions of the field area. The figure below compares the three Scaled to fit formats.


Scaled to fit


Scaled to fit (proportional)


Scaled to fit centered (proportional)

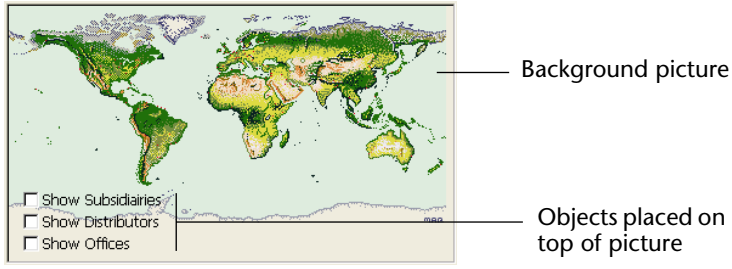
When you use Scaled to fit (proportional), the picture is reduced proportionally on all sides to fit the area created for the picture. The Scaled to fit centered (proportional) option does the same, but centers the picture in the picture area.

If the picture is smaller than the area set in the form, it will not be modified.

If the picture is bigger than the area set in the form, it is proportionally reduced. Since it is proportionally reduced, the picture will not appear distorted.

On Background

On Background makes the picture transparent. Any objects placed behind the graphic such as fields or variables are visible through the graphic. When a Picture field is in this format, the user can move the picture around the inside of the Picture field by dragging it. 4D remembers the object's position on the background. The figure below shows a form that includes a picture with the On Background format.

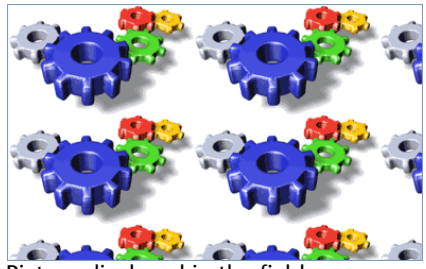


Note If you are printing pictures with the On Background format, they will be printed as bitmaps.

Replicated

When the area that contains a picture with the Replicated format is enlarged, the picture is not deformed but is replicated as many times as necessary in order to fill the area entirely.

Original picture



Picture displayed in the field

If the field is reduced to a size smaller than that of the original picture, the picture is truncated (non-centered).

Managing Active Objects on a Form

This section describes the active objects you can use on a form to control database and interface functions. Active object s include buttons, radio buttons, check boxes, tab controls, pop-up menus, drop-down lists, hierarchical pop-up menus and hierarchical lists, combo boxes, list boxes, scrollable areas, splitters, gauges, and plug-ins. Enterable objects (variables) are treated much the same as fields and are described in the previous section.

This section explains how to modify active objects other than fields, enterable objects or subforms. The different active object types are discussed in detail in the section “Types of Active Objects” on page 549.

Active Object Properties

An object’s properties can be modified using the Property List. Certain properties can be accessed using the contextual menu (**right-click**) or the **Object** menu of the Form editor.

For more information about using the Property List, refer to “The Property List” paragraph on page 388.

You can select several objects to edit their common properties.

Active object properties include, for instance, its name, type, action, drag-drop properties, movement options, appearance and associated method.

The following are the characteristic properties of active objects:

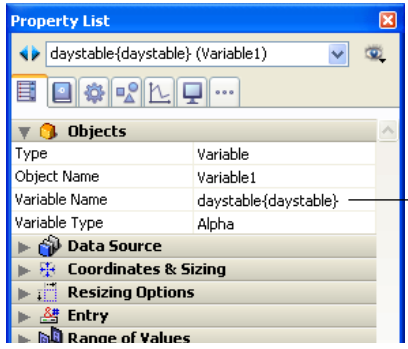
- **Object** and **Variable**: Each active form object is associated with a variable[1]. The variable name can be different from the object's name. When using 4D's language, you can refer to an active form object by either its variable name or object name. For more information about naming rules for variables, please refer to the "Rules for Naming Tables and Fields" paragraph on page 287 (the rules are identical). Refer to the "Identifiers" section of the 4D *Language Reference* manual for more information about the different types of variables.
  You can set the object type and, for certain objects, the type of the variable associated with it.
- **Data Source**: The contents of certain objects (tabs, for instance) can be set via default values or choice lists.
- **Coordinates & Sizing**: You can set the size, location, resizing options, and positioning options for each object.
- **Entry**: Enterable objects have properties related to their activation using the keyboard (for example, the possibility of visually displaying the focus) as well as entry filters.
- **Display**: This theme includes the **Invisible by Default** property and, for enterable objects, the display format.
- **Appearance, Background, Border, Colors**, **Text**: You can set the appearance of each object individually (in particular its color, border line style and platform interface), as well as, for objects containing text, the font, color, size and font style, alignment and style sheet.
- **Action**: These properties are used to set standard actions or methods associated with the object, as well as its behavior during drag-and-drop operations.
- **Events**: You can set a list of events for which the object's method will be executed.
- **Help**: You can associate a help tip with each active object of the form.

1. It is possible to use an expression instead of a variable name. For more information, please refer to the following paragraph "Using Expressions as Variable Names" on page 539.

Keep in mind that several types of objects have specific properties that are used to configure their behavior: objects using pictures (such as picture buttons or picture pop-up menus), 3D buttons, subforms, etc. These properties are described in paragraphs concerning these objects.

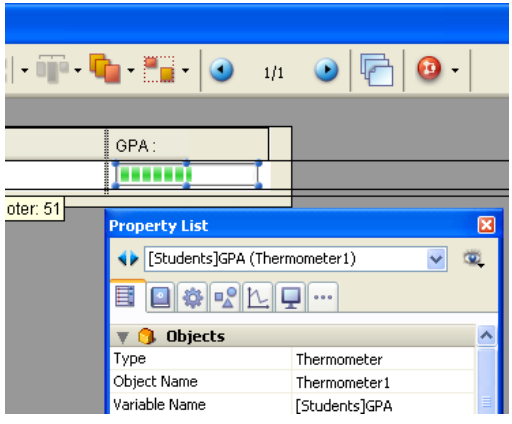
Using Expressions as Variable Names

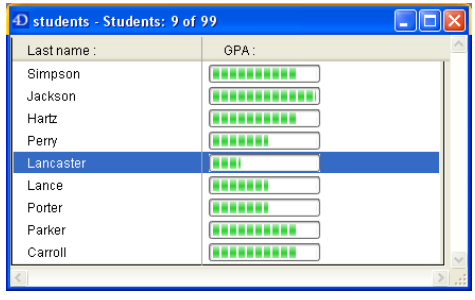
Variables associated with form objects can now contain any type of expression that returns a value — and no longer variable names only. You can simply enter the expression directly in the "Variable Name" area of the Property List for the object:



Any valid 4D expression is allowed: simple expression, formula, 4D function, project method name or field using the standard *[Table]Field* syntax. The expression is evaluated when the form is executed. Naturally in this case it is not possible to manage the value of the variable by programming.

Multiple possibilities are offered by this principle. For example, it means that you can associate a numeric field with a thermometer in order to display values graphically in lists:





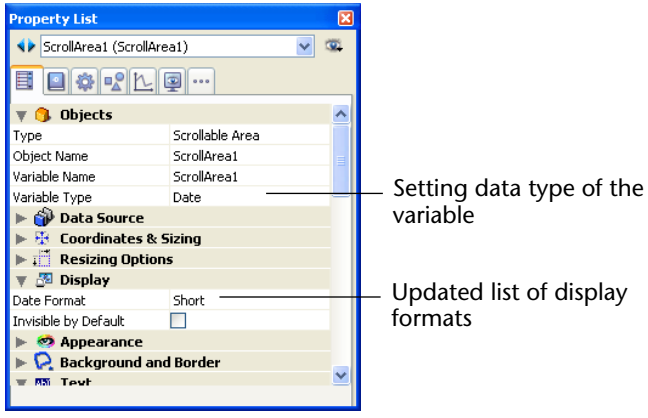
Creating an Active Object

Active objects can be created in a form in one of the following ways:

- With the object bar of the Form editor by selecting one of the objects then drawing it in the form. Some objects include variants. For more information about the contents and operation of the object bar, refer to the "Form Editor Object Bar" paragraph on page 380. For a detailed description of active object types, refer to the "Types of Active Objects" paragraph on page 549.
- By pasting or duplicating an object of the same form, or by pasting an object from another form. The name as well as the properties (if applicable) of the object are automatically adapted by 4D.
- By pasting or dropping an object from an object library (for more information about this point, refer to the "Using Object Libraries" paragraph on page 461).

Display Formats for Objects

The display formats provided for enterable and non-enterable active objects are identical to those provided for fields. However, you must set the type of data that will be processed by the variable. The default type is Alpha. This option can be used to update the list of display formats:



For more information about display formats, refer to the "Display Formats" paragraph on page 521.

Data Entry Controls for Enterable Objects

The Property List window provides data entry controls for enterable objects as well as for fields. These controls allow you to:

- Set an entry filter that defines the characters to be allowed,
- Display a choice list,
- Establish lists of required values or excluded values,
- Set maximum and minimum allowable values,
- Set default values or default lists of values.

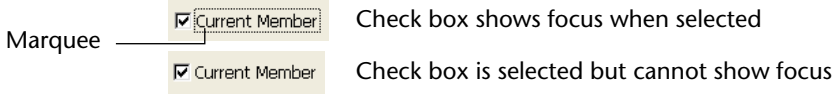
These controls work for enterable objects exactly as they work for fields. For complete information, see the appropriate sections earlier in this chapter.

Setting the Tabable and Focusable Properties

The Tabable and Focusable properties can be set in the "Entry" theme of the Property List. These two options will affect how the object appears during data entry.

- When the Focusable property is selected for an object, the object can have the focus (and can thus be activated by the keyboard for instance). It is outlined by a gray dotted line when it is selected — except when the "Hide focus rectangle" option has also been selected (see the "Hide Focus Rectangle" paragraph on page 491.
- If an object has the Tabable property, it is included in the data entry order. In other words, the user can press the Tab key to select the object.

  The Tabable property is only accessible if the Focusable property is selected. In other words, any tabable object can have the focus. However, some objects can be "focusable" while not being "tabable" (for example, an object can be selected by clicking it and not "tabable"). In this case, the object does not belong to the data entry sequence.



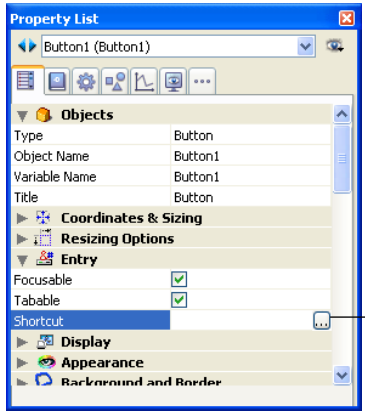
Note A non-enterable variable cannot be "tabable."

Assigning a Keyboard Shortcut

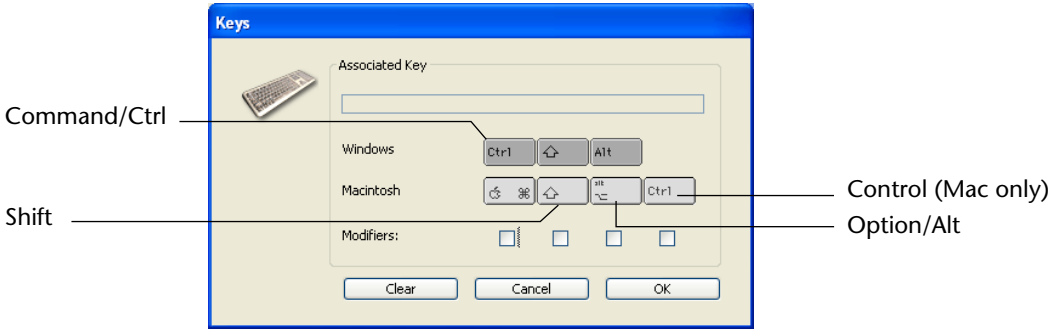
You can assign a keyboard shortcut for buttons and check boxes. The user can then activate the button or select the check box using the keyboard instead of having to use the mouse.

► To assign a keyboard shortcut to an object:

1 **In the Property List, click the [...] button in the Shortcuts property located in the "Entry" theme[1].**



The following dialog box appears.



2 **Type the keyboard shortcut.**

For example, if you want to use Ctrl+H, hold down the **Ctrl** key and press H. The letter H will then appear in the Associated Key area and the check box below the Ctrl key will be checked.

You are not required to use modifier keys. You can use any key alone as the shortcut, although this is not recommended in most cases.

---

1. You may need to click in the Shortcut entry area of the Property List to make the button visible.

If you like, you can manually modify the selection of modifier keys by selecting or deselecting any of the modifier key check boxes.

To start over again, click **Clear**.

3 **When you have finished, click OK.**

The Property List displays the keyboard shortcut that was assigned to the object.
If you want to change the shortcut later, simply open the Shortcuts dialog box and type the key combination you want to use.

4D displays the new combination in the dialog box.

Enabling Drag and Drop

Active objects on a form may have drag-and-drop properties. Two options located in the "Action" theme of the Property List, are related to drag and drop.

- **Draggable**: Controls whether the user has the ability to drag the object,
- **Droppable**: Controls whether the object has the ability to "receive" a dragged object.

If you want to enable drag and/or drop for a particular object, enable the appropriate property. You then must manage the drag-and-drop action using a method. For more information, see the section on Drag and Drop commands in the *4D Language Reference* manual.

Automatic Drag and Drop

Active text areas (fields and variables), combo boxes and picture areas (fields and variables) include additional options intended for managing automatic drag and drop. Automatic drag and drop can be used to copy or move text or pictures directly in form areas. It can be used in the same 4D area, between two 4D areas, or between 4D and another application (for example, WordPad®).

For example, automatic drag and drop lets you copy a value between two fields without using programming:

When automatic drag and drop is activated, the internal 4D drag-and-drop management mechanisms are not used. If you wish to force the use of the internal 4D drag and drop with active text areas, hold down the **Alt** (Windows) or **Option** (Mac OS) key before performing the drag and drop.

- **Automatic Drag**: This property activates the automatic drag mode for text areas. This mode takes priority, even when the **Draggable** option is checked.

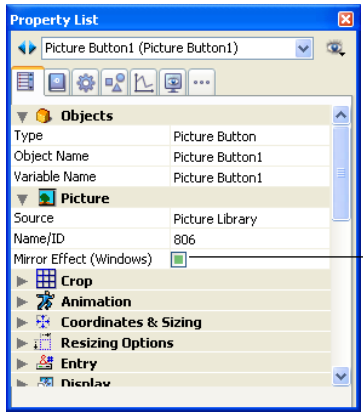
Note Automatic drag moves the text selection. To copy the selection, hold down the **Ctrl** (Windows) or **Command** (Mac OS) key during the action.

- **Automatic Drop**: This property activates the automatic drop mode. In this mode, 4D automatically manages — if possible — the insertion of dragged data of the text or picture type that is dropped onto the object (the data are pasted into the object). The On Drag Over and On Drop form events are not generated in this case.
  In the case of data other than text or pictures (another 4D object, file, etc.) or complex data being dropped, the application refers to the value of the **Droppable** option: if it is checked, the On Drag Over and On Drop form events are generated; otherwise, the drop is refused. This also depends on the value of the "Prevent drop of data not coming from 4D" option (see the "Design Compatibility" paragraph on page 173).

  For more information about drag and drop in 4D, please refer to the "Drag and Drop" chapter of the *4D Language Reference* manual.

Mirror Effect (Windows)

The **Mirror Effect (Windows)** option (found in the "Picture" theme of the Property List) can be applied to any active objects using pictures:



This option causes the picture associated with the object to be inverted when it is displayed within an application configured in "right-to-left" mode under Windows. It is available for picture fields and variables, Picture Buttons and Picture Pop-up Menus, 3D Buttons, 3D Check Boxes and 3D Radio Buttons, picture radio buttons, as well as for static pictures.
This option is detailed in the "Mirror Effect for Pictures (Windows)" paragraph on page 460.

Duplicating on a Matrix

Sometimes you may want to place several similar active objects in a form at the same time, numbering them sequentially so that their names are unique. For example, you may want to create a series of buttons that perform database operations. Duplicating on a matrix has the additional advantage of quickly and easily aligning multiple objects.

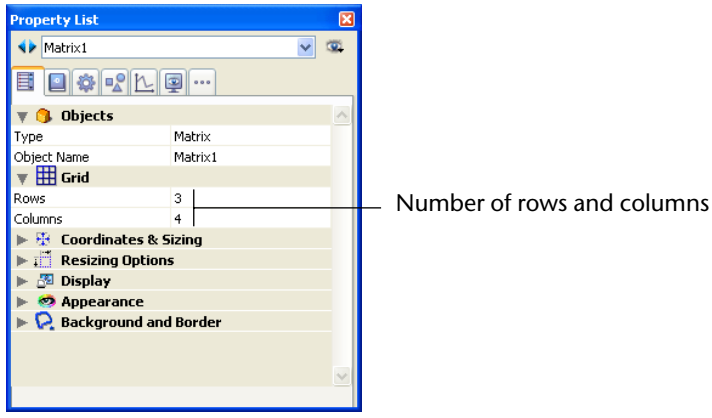
You can either duplicate an active object on a matrix manually or use the Duplicate Many dialog box, which allows you to quickly populate the matrix.

► To duplicate an object on a matrix:

1 **Select the Matrix tool**  **in the object bar (variation of the Rectangle group) and create a matrix on the form.**
Make sure that each cell in the matrix is large enough to contain the object you want to duplicate.

2 **Display the Property List and select the matrix you just created.**

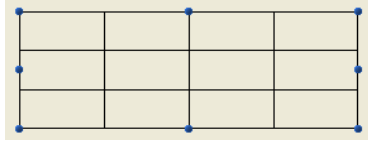
3 **In the "Grid" theme, set the number of rows and columns of the Matrix.**



4 **If necessary, set the appearance of the matrix using the options in the Appearance theme.**

You can set the border style, the line thickness, the color, and the fill pattern.

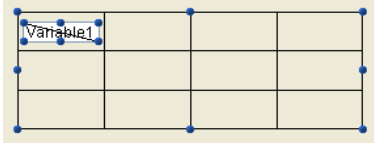
5 **Make sure the cells in the matrix are large enough to contain the object that you want to duplicate.**



6 **Create a new active object and place it in the upper left cell of the matrix.**

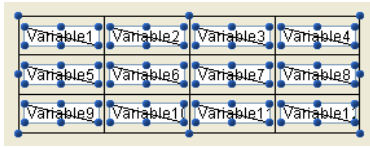
Make sure that the object type matches the type and size of the object you want to duplicate.

7 **Select both the object and the matrix.**



8 **Choose Duplicate on Matrix from the Object menu.**

4D copies the active object into each cell in the matrix, giving each active object a unique number.



The objects are copied along with their size and style properties as well as their associated method (where applicable).
4D numbers the active objects from top to bottom in each column. These numbers are added to the object name for each object, thus creating a unique object in every matrix cell.

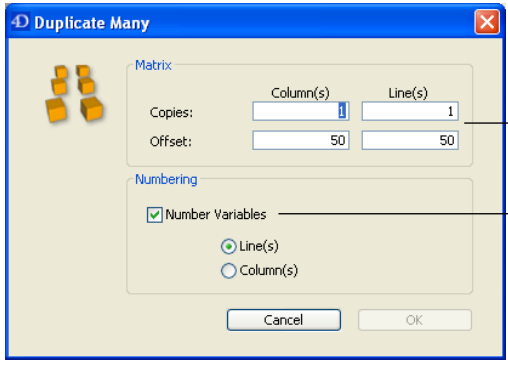
Note To number the series of active objects from left to right in each row, hold down the **Alt** key (Windows) or **Option** key (Mac OS) when you choose **Duplicate on Matrix**.

You can refer to these objects in methods using the names they have been given. You can delete the matrix or leave it in the form.

► To duplicate one or several objects using the Duplicate Many dialog box:

1 **Select the object(s) you want to duplicate.**

2 **Select Duplicate Many from the Object menu.**



3 **In the Matrix area, enter the number of lines and columns as well as the offset between them.**

For more information about this point, refer to "Duplicating Objects" on page 427.

4 **Select the "Number Variables" option.**

This option is enabled only if you selected a variable.

5 **Select the order of the numbering.**

If you select the **Line(s)** option, 4D will number active objects from left to right starting from the top towards the bottom (line by line).
If you select the **Column(s)** option, 4D will number active objects from top to bottom starting from the left towards the right (column by column).

6 **Click the OK button.**

The objects are copied and numbered according to your settings:

Example using the Line(s) numbering option

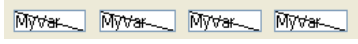
Incrementing a Set of Variables

4D provides a handy shortcut for automatically incrementing a group of variables that has been created in a form. This shortcut can be used, for example, to reorganize the variables for the buttons of a form or to make sure that each form variable is unique. Keep in mind that this function only modifies the names of variables associated with objects, not the names of the objects themselves.

► To automatically increment a set of variables:

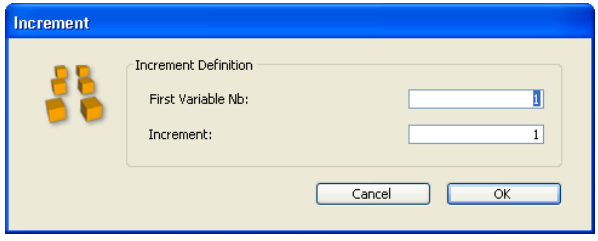
1 **Select each active object whose variable needs to be renumbered.**

You can select any kind of active object.



2 **Under Windows, Ctrl+Alt+click on one of the objects.**
Under Mac OS, Command+Option+click on one of the objects.

The following dialog box appears:

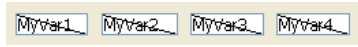


Note This dialog box will not appear if the selection contains an object that is not active or a field.

3 **Set the start number and the increment to be applied.**

4 **Click on OK.**

The variables of the selected objects are immediately renumbered using the parameters set. The numbers are added to the end of the current variable names. Renumbering is carried out from left to right and top to bottom.



Types of Active Objects

4D provides the following types of active objects (in addition to fields):

- Variables (enterable and non-enterable) of the Alpha, Text, Numeric, Date, Time and Picture type,
- Buttons
- 3D Buttons
- Highlight buttons
- Invisible buttons
- Picture buttons
- Button grids
- Check boxes
- 3D Check boxes
- Radio buttons
- 3D Radio buttons
- Picture radio buttons
- Scrollable areas
- Hierarchical lists
- List boxes
- Combo boxes
- Pop-up menus/Drop-down lists
- Hierarchical menus
- Picture pop-up menus

- Tab controls
- Plug-in areas
- Button grids
- Thermometers
- Indicators
- Rulers
- Splitters.

The sections that follow describe each kind of object in detail.

Enterable and Non-enterable Variables

An enterable variable allows the user to enter a value into a variable and to display this value. A non-enterable variable allows you to display the value of a variable without it being modifiable by the user. You use methods to manage enterable and non-enterable variables.

Variables are used for temporary storage of data. One common use for a variable is to display calculations that are done using a method such as:

```
vTotal:= Quantity * Price
```

You create a variable that displays the result of the calculation, name this variable vTotal, and use a method to do the calculation.

An enterable variable accepts data. You can set data entry controls for the object as you would for a field. The entered data is associated with the object name. You can manage the data with object or form methods using the object's name as a variable.

A non-enterable variable only displays data. The displayed data is associated with the name of the variable. You control the data with methods, using the name of the variable. The form events On Clicked and On Double Clicked can also be used with non-enterable objects. This makes managing customized contextual menus easier.

To create a non-enterable variable, create a standard variable and deselect the **Enterable** property. This property can be set in the "Entry" theme of the Property List.

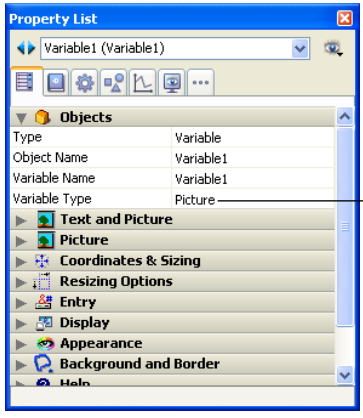
Enterable and non-enterable variables can be of any size. When they display characters, the size of the variable area varies in steps related to the font size of the characters used. Enterable and non-enterable variables can make use of display formats. Text and picture objects can use horizontal and vertical scroll bars and can be printed with a variable frame (see the "Printing Text Fields" paragraph on page 679).

Typing of Picture Variables in Forms

Specific native mechanisms govern the display of picture variables in forms. These mechanisms require greater precision when configuring variables: from now on, they must have already been declared *before* loading the form — i.e., even before the On Load form event — unlike other types of variables.

To do this, you simply need:

- Either for the statement C_PICTURE(varName) to have been executed before loading the form (typically, in the method calling the DIALOG command),
- Or for the variable to have been typed at the form level using the **Variable Type** pop-up menu of the Property List:



Otherwise, the picture variable will not be displayed correctly (only in interpreted mode).

Buttons

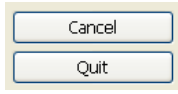
The Form editor lets you add a wide variety of buttons to your forms. When you add buttons to a form, you can associate a standard action with each button. Automatic buttons let the user accept, cancel, or delete records, move between records, move from page to page in a multi-page form, and open, delete, or add records in a subform, etc.

You normally add buttons when you create the form using the Form Wizard. You can modify these buttons' actions in the Property List. For example, you can remove a standard action from a button and write an object method that specifies the button's action.

You can also add buttons and assign button actions with the Form editor. For example, if you need more than one subform on the form, you can add the necessary additional subforms and automatic buttons in the Form editor. You simply add each button to the form and associate a standard action with each one.

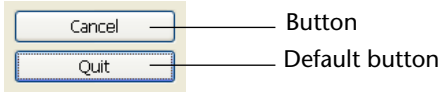
4D lets you use the following types of buttons:

- **Buttons**: These buttons are displayed in the current platform interface. Button text is displayed in the selected font, font size, style, and color.



The label displayed by the button is set in the **Title** field of the "Objects" theme in the Property List. You can change it at any time.

- **Default Buttons**: A default button looks exactly like a standard button, except that its border is thicker. This indicates to the user that the button is the recommended choice.
The following illustration compares a button to a default button.



Under Mac OS, the default buttons are pulsing buttons:

The **Default button** object type does not exist as such but it is a property that is available for buttons.

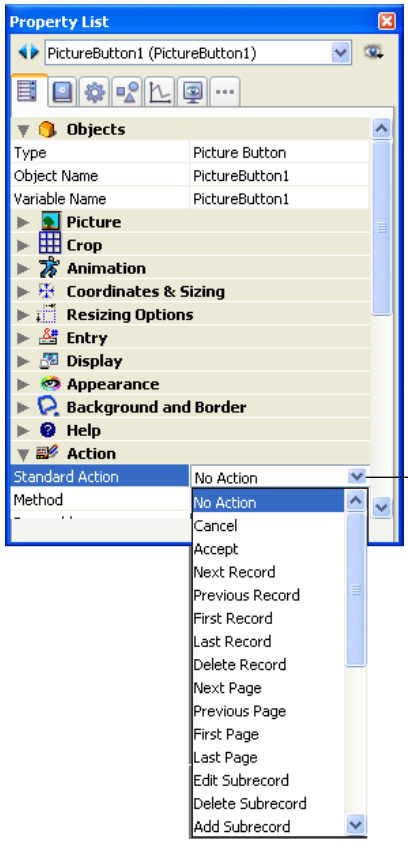
Note There can only be one Default button per form page.

- **Highlight buttons and Invisible buttons**: These buttons are designed to be placed on top of graphic objects.
  Highlight buttons are transparent. When the user clicks a highlight button, the graphics of the button are highlighted.
  Invisible buttons remain invisible and do not highlight when clicked. The resulting action, such as displaying a different page, should indicate that the button has been clicked. An invisible button should be placed on top of text or a graphic that denotes its function; the user clicks on the text or graphic, and the button is activated.
- **3D buttons** and **Picture buttons**: The family of 3D buttons (3D buttons, 3D check boxes, and 3D radio buttons) and picture buttons include numerous specific properties. These buttons are described in the "3D Buttons, 3D Check Boxes and 3D Radio Buttons" paragraph on page 558 and the "Picture Buttons" paragraph on page 566.

Button Activation Management

Buttons are dimmed when appropriate during form execution. For example, if the first record of a table were displayed, the First Record button would appear dimmed.
You create a button by choosing the desired button type from the Type drop-down list.

You then choose the standard button action you want from the Standard Action drop-down list.



Display of standard actions list

Note You can also use the **Standard Action** hierarchical command in the contextual menu of the editor (**right-click** on the object).

Regardless of the type of button, you must select an item from this drop-down list. If you want the button to perform an action not listed in the Standard Action drop-down list, choose “No Action” and write a method that specifies the button’s action. Normally, you would activate the On Clicked event in the Events theme and the method would run only when the button is clicked.

All variables associated with buttons (including regular buttons, highlight buttons, invisible buttons, radio buttons, picture radio executed for the first time in Design or Application mode. When the user clicks a button, its variable is set to 1. You can associate a method with any button.

Standard Button Actions

This section discusses each standard action that can be assigned to a button (or to menu commands, see the "Associating a Standard Action" paragraph on page 878).

- **No Action**: Use a **No Action** button for a button that does not perform a standard action. Choose **No Action** when you need to write a method to manage the button. For example, a button that displays a custom Find dialog box in a custom application would have a **No Action** standard action because you must write a method to open the custom dialog box.
- **Accept** and **Cancel Actions**: Clicking an **Accept** button saves a record. It triggers an On Validate event. Clicking a **Cancel** button exits the current record without saving any changes.
- **Delete Record Action**: Clicking a **Delete Record** button displays an alert asking the user to confirm the deletion. Clicking Yes in the alert deletes the current record. If the user is using the input form for a subrecord, **Delete Record** deletes the current subrecord.

  After the user clicks a **Delete Record** button, 4D automatically returns to the output display.

Note A button with this action is automatically disabled when a new record is being added.

- **Record Navigation Actions**: The **Next Record**, **Previous Record**, **First Record**, and **Last Record** buttons first accept the current record and then make the specified record current. The specific record made current by these buttons depends on the sort order.

  These buttons perform the appropriate actions for subrecords when the user is entering subrecords.

  A button of this type is automatically disabled when its action is inappropriate. For example, if the user displays the first record, the **Previous Record** button would be disabled.
- **Page Navigation Actions**: The **First Page**, **Last Page**, **Next Page**, and **Previous Page** buttons display the appropriate page in a multi-page form. If there is only one page, these buttons are inactive.

  A button of this type is automatically disabled when its action is inappropriate. For example, if the user displays the first page, the **Previous Page** button would be disabled.

- **Edit Subrecord**: This action can be used in two contexts: subforms and list forms displayed using the MODIFY SELECTION / DISPLAY SELECTION commands or in the records display window in Design mode.
  - *Subform*: Placed in the form of a parent record, this button is active when a record of the subform is selected. If the user selects a subrecord and then clicks the **Edit Subrecord** button, the subrecord switches to editing mode, either directly in the list, or in the associated detail form (depending on the properties of the subform).
  - *List forms*: This button is active when a record is selected in the list. When a user clicks this button, the record switches to editing mode. With lists displayed via the MODIFY SELECTION / DISPLAY SELECTION commands, the modification is carried out in the list or on the detail page depending on the value of the *enterList* parameter. In the records display window, the modification is carried out on the detail page (the action is equivalent to a double-click).
- **Add Subrecord**: This action can be used in two contexts: subforms and list forms displayed using the MODIFY SELECTION / DISPLAY SELECTION commands or in the records display window in Design mode.
  - *Subform*: Placed in the form of a parent record, this button is active when a record of the subform is selected. When the user clicks this button, 4D creates a new record in the table or related table, either directly in the list, or in the associated detail form (depending on the properties of the subform).
  - *List forms*: This button is always active. When the user clicks this button, a new blank record is created. With lists displayed using the MODIFY SELECTION / DISPLAY SELECTION commands, the record is added in the list or in the detail page depending on the value of the *enterList* parameter. In the records display window, the record is added to the list.
- **Delete Subrecord**: This action can be used in two contexts: subforms and list forms displayed using the MODIFY SELECTION / DISPLAY SELECTION commands or in the records display window in Design mode.
  - *Subform*: Placed in the form of a parent record, this button is active when a subrecord has been selected in a subform.
  - *List forms*: This button is active when at least one record is selected in the list.

When the user clicks this button, a dialog box, which can be used to confirm or cancel the deletion, appears.

For more information about using buttons in subforms, refer to the "Adding a Subform to the Form" paragraph on page 621.

- **Automatic Splitter**: This standard action allows you to create custom splitters on a form.

  This action can only be assigned to an invisible button. When an invisible button is assigned this action, it behaves in the same way as a splitter. By pasting for example a picture in the invisible button, you can create any type of custom interface for your splitters.

  For more information about splitters, refer to "Splitters" on page 589.

- **Undo**: Cancels the last action performed (=**Undo** command of the **Edit** menu). *Undo* should not be confused with *Cancel* (= cancels any modifications made to a record during its viewing and returns to the Output form).
- **Redo**: Repeats the last action cancelled (= **Redo** command of the **Edit** menu).
- **Cut**: Removes the selection and places it in the Clipboard.
- **Copy**: Places a copy of the selection in the Clipboard.
- **Paste**: Inserts the contents of the Clipboard at the location of the insertion point.
- **Clear**: Deletes the selection. If nothing is selected, it erases the entire area containing the cursor (enterable areas only).
- **Select All**: Selects all of the selectable elements in the context.
- **Show Clipboard**: Opens a new window that displays the current contents of the Clipboard.
- **Preferences**: Displays the standard 4D application Preferences dialog box (see note below).
- **Application**: Brings the windows and menu bars of the 4D Application environment to the foreground.

- **Quit**: Displays an "Are you sure?" confirmation dialog box, then exits the 4D application if validation occurs. Otherwise, the operation is cancelled.
  When this action is assigned to a button with which an object method is also associated, the following sequence is executed: first, the confirmation dialog box appears. If it is validated, 4D executes the object method. After its execution, the application quits.

Note Mac OS Under Mac OS X, the custom menu commands associated with the **Preferences** and **Quit** actions are automatically placed in the application system menu, when the database is running in this environment. This mechanism simplifies the management of the **Quit** command under Mac OS X

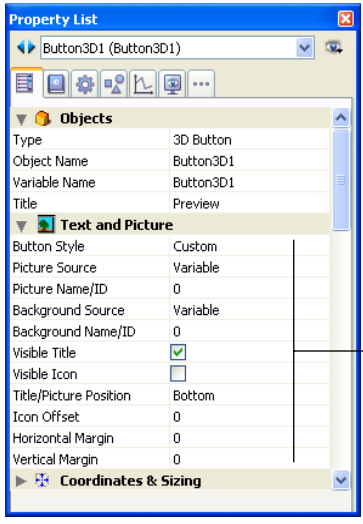
- **Return to Design mode**: Brings the windows and menu bars of the 4D Design environment to the foreground.
  When the database is running in interpreted mode, this displays the current window of the Design environment. When the database is running in compiled mode, this displays the records window of the current table (in compiled mode, only access to records is possible).
- **MSC**: Displays the Maintenance and Security Center window.

3D Buttons, 3D Check Boxes and 3D Radio Buttons

The family of 3D buttons includes **3D buttons**, **3D check boxes** and **3D radio buttons**. These objects are structurally identical; the only difference is the processing of the associated variable:

- The associated variable of a button equals 0 when the form is opened (standard state); it equals 1 when the user clicks the button (pressed state); then returns to its standard state and equals 0 again.
- The associated variable of a 3D check box equals 0 when the box is not checked and is set to 1 when the box is checked. Unlike buttons, the 3D check box keeps its state (0 or 1) until the user selects it again. For more information, refer to the "Check Boxes" paragraph on page 572.
- 3D radio buttons operate in groups; the associated variable of the selected button equals 1 and the others equal 0. The variable can equal either 0 or 1 when the form is opened. For more information, refer to the "Radio Buttons and Picture Radio Buttons" paragraph on page 574.

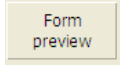
The family of 3D buttons offers numerous specific properties, which can be used to set up interfaces that can be perfectly integrated into different operating systems. More particularly, it is possible to apply different pre-defined styles to 3D buttons (bevel buttons, push buttons, etc.) or to associate pop-up menus with them.
A great number of variations can be obtained by combining the various properties.



3D button properties (Custom style)

Title

This property allows inserting a label in the button. The font and the style of this label can be set in the “Text” theme.
You can force a carriage return in the label by using the \ character (backslash).



To insert a \ in the label, enter \\.

By default, the label is placed in the center of the button. When the button also contains an icon, you can modify the relative location of these two elements using the **Title/Picture Position** property.
It is possible to hide the title by deselecting the **Visible Title** property. In this case, the icon is automatically moved to the center of the button.

For the purpose of database translation, you can enter an XLIFF or STR# reference in the title area of a button (see Appendix C on page 1293).

Button Style

This property allows setting the general appearance of the button. The style also plays a part in the availability of certain options. The following styles are available:

- **None** (default value)

Windows

Mac OS

A 3D button with the “None” style is similar to an invisible button: its “highlight” is not displayed. However, it takes advantage of the 3D button options.
Under Mac OS, it is not possible to display the triangle indicating a linked pop-up menu.

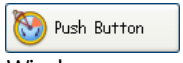
- **Background Offset**


Windows

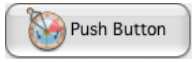

Mac OS

This style corresponds to highlight buttons, except that when the user clicks on this type of button, a 3D effect is obtained by offsetting the picture located under the button.
When the “Background Offset” style is selected, no options are available.

- **Push Button**



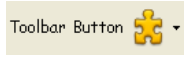
Windows

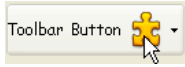


Mac OS

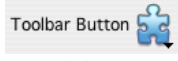
A 3D button with the “Push Button” style appears as a standard system button. It nevertheless benefits from 3D button options, with the exception of the “With Pop-up Menu” property.

- **Toolbar Button**





Windows

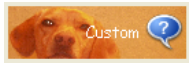


Mac OS

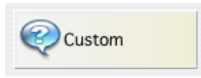
This style of 3D button is primarily intended for integration in a toolbar.

Under Windows, its highlight appears when the mouse rolls over it. When it uses the “With Pop-up Menu” property, a triangle is displayed to the right and in the center of the button.
Under Mac OS, the highlight of the button never appears. When it uses the “With Pop-up Menu” property, a triangle is displayed to the right and at the bottom of the button.

- **Custom**



Windows

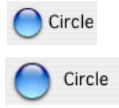


Mac OS

This 3D button style accepts a custom background picture and allows managing various additional parameters (icon and margin offset). For more information, please refer to the “Custom 3D Buttons” paragraph on page 565.
This style benefits from 3D button options, with the exception of the “With Pop-up Menu” property.

- **Circle**

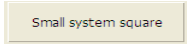
Windows



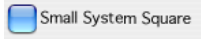
Mac OS

Under Mac OS X, a 3D button with the “Circle” style appears as a round system button. Two set styles are available for the circle, which depend on the size of the button in the form.
This style benefits from 3D button options, with the exception of the “With Pop-up Menu” property.
Under Windows, this button style is identical to the “None” style (the circle in the background is not taken into account).

- **Small System Square**



Windows

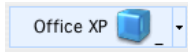


Mac OS

Under Mac OS X, a 3D button with the “Small System Square” style appears as a system square. This style benefits from 3D button options, with the exception of the “With Pop-up Menu” property.
Under Windows, this button style is identical to the “None” style (the square in the background is not taken into account).

- **Office XP**

Windows



Mac OS

A 3D button with the “Office XP” style has the following characteristics:

- The colors of its highlight and background are based on the system colors,
- Under Windows, in use, its highlight only appears when the mouse rolls over it.

This style benefits from 3D button options.

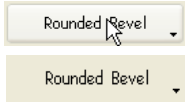
- **Bevel**

Windows

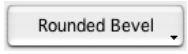
Mac OS

Under Mac OS, a “Bevel” button appears as a standard system button. It benefits from 3D button options, including the “With Pop-up Menu” property.
Under Windows, this button style is similar to the “Toolbar button” style, the only difference being that the triangle indicating the presence of an associated pop-up menu is located at the bottom right of the button.

- **Rounded Bevel**



Windows



Mac OS

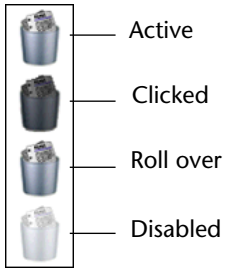
Under Mac OS, a “Rounded bevel” button is similar to a “Bevel” button except that its highlights are rounded.
Under Windows, this button style is identical to the “Bevel” style.

Adding an Icon to a 3D Button

It is possible to add an icon to any 3D button style (with the exception of the “Background Offset” style):

The picture used must contain four distinct vertical areas, which will be used by 4D to represent the four standard states of the button: **active**, **clicked**, **roll over** and **disabled**.

For example, here is a picture used for 4D forms:



Managing associated icons is done using several properties such as Picture Source and Picture Name/ID.

- **Picture Source**
  Like with picture buttons, the icon of a 3D button can come from four different sources: **Variable**, **Picture Library**, **Resource File** or **File**. Once the source is set, you can indicate the name or the number of the picture in the “Picture Name/ID” property.

- **Picture Name/ID**
  Once the picture source is set, enter the name (if the picture is a variable or comes from the picture library), the number (if the picture comes from the picture library or a resource file) or the pathname (if the picture comes from a picture file) of the picture in this area; in the case of a picture file, the pathname must be relative to the Resources folder of the database (see the “Automatic Referencing of Picture Files” paragraph on page 456).
  Pass 0 in this property if you do not wish to add a 3D button picture.

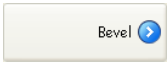
Note You can associate a picture with a button by dragging and dropping a picture from the picture library or from a disk file.

- **Visible Title / Visible Icon**
  When the button includes both a title and a picture, you can hide one or the other item by deselecting the **Visible Title** or **Visible Icon** option (or both). When one of these items is hidden, it is automatically replaced by the other in the center of the button.

Title/Picture Position

This property allows modifying the relative location of the button title in relation to the associated icon. This property has no effect when the button contains only a title (no associated picture) or a picture (no title).
By default, when a 3D button contains a title and a picture, the text is placed below the picture.

Here is the result of the various options for this property:



- **Left**: The text is placed to the left of the icon. The contents of the button are aligned to the right.

- **Top**: The text is placed above the icon. The contents of the button are centered.

- **Right**: The text is placed to the right of the icon. The contents of the button are aligned to the left.

- **Bottom**: The text is placed below the icon. The contents of the button are centered.

- **Centered**: The text of the icon is centered vertically and horizontally in the button. This parameter is useful, for example, for text included in an icon.

With a Pop-up Menu

This property allows displaying a symbol that appears as a triangle in the 3D button, which indicates that there is a pop-up menu attached:

The appearance and location of this symbol depend on the button style and the current platform. 3D button styles that accept the “With Pop-up Menu” are the following:

- None
- Toolbar Button
- Bevel
- Rounded Bevel
- Office XP

- **Linked and Separated**
  To attach a pop-up menu symbol to a 3D button, there are two display options available: **Linked** and **Separated**.

Separated Linked

Note The real availability of a "separated" mode depends on the style of the button and the platform.

Each option specifies the relation between the button and the attached pop-up menu:

- When the pop-up menu is *separated*, clicking on the left part of the button directly executes the current action of the button; this action can be modified using the pop-up menu accessible in the right part of the button,
- When the pop-up menu is *linked*, a simple click on the button only displays the pop-up menu. Only the selection of the action in the pop-up menu causes its execution.

These options also influence the handling of form events by the button (for more information on this, refer to the *4D Language Reference* manual).

- **Managing the pop-up menu**
  It is important to note that the "With Pop-up Menu" property only manages the graphic aspect of the button. The display of the pop-up menu and its values must be handled entirely by the developer, more particularly using form events and the Dynamic pop up menu and Pop up menu commands.

Custom 3D Buttons

When the "Custom" 3D button style is selected, several additional properties are available: **Background Source**, **Background Name/ID**, **Icon Offset**, **Horizontal Margin** and **Vertical Margin**.

- **Background Source**: This property allows you to set the picture that will be drawn in the background of the button. Like with the icon, you can indicate whether the picture comes from a variable, picture library, resource file or disk file.
- **Background Name/ID**: Once the source is set, you can indicate the name or number of the picture in this area.

Like with icons, the background pictures can contain four distinct vertical areas, which will be used by 4D to represent the four standard states of the button: **active**, **clicked**, **roll over**, and **disabled**. Note that the effect produced during a click can also be handled using the **Icon Offset** property.

- **Icon Offset**
  This property allows setting a custom offset value in pixels, which will be used when the button is clicked: the title of the button will be shifted to the right and toward the bottom for the number of pixels entered. This function allows applying a customized 3D effect when the button is clicked.

- **Horizontal Margin** / **Vertical Margin**
  These properties allow you to set the size (in pixels) of the internal margins of the button. These margins delimit the area that the 3D button icon and title must not surpass.
  This parameter is useful, for example, when the background picture contains borders:

Custom 3D button without margin

3D button with 13-pixel margin

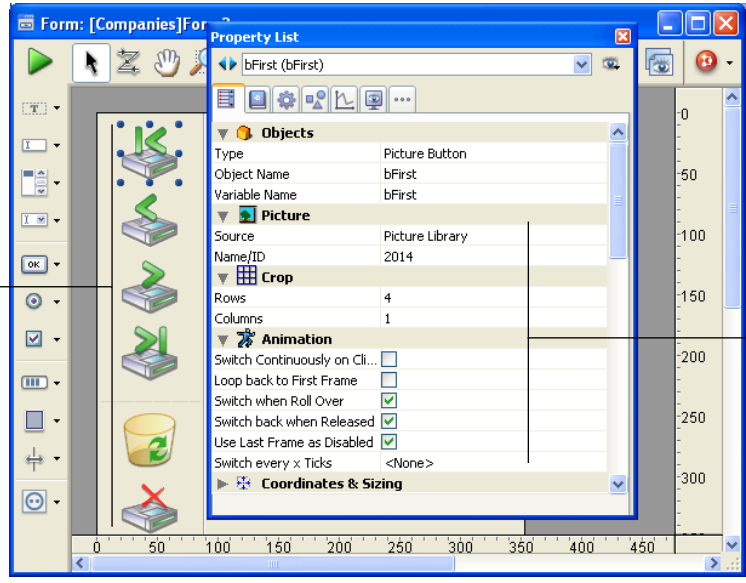
Picture Buttons

A picture button can have several different states— by comparison, a standard button accepts three states: enabled, disabled and clicked. As the name indicates, with a picture button each state is represented by a different picture.

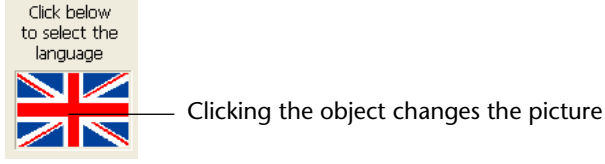
Picture buttons can be used in two ways:

- As command buttons in a form. In this case, the picture button generally includes four different states: enabled, disabled, clicked and rolled over.

This is what is used by the Form Wizard for most form templates:

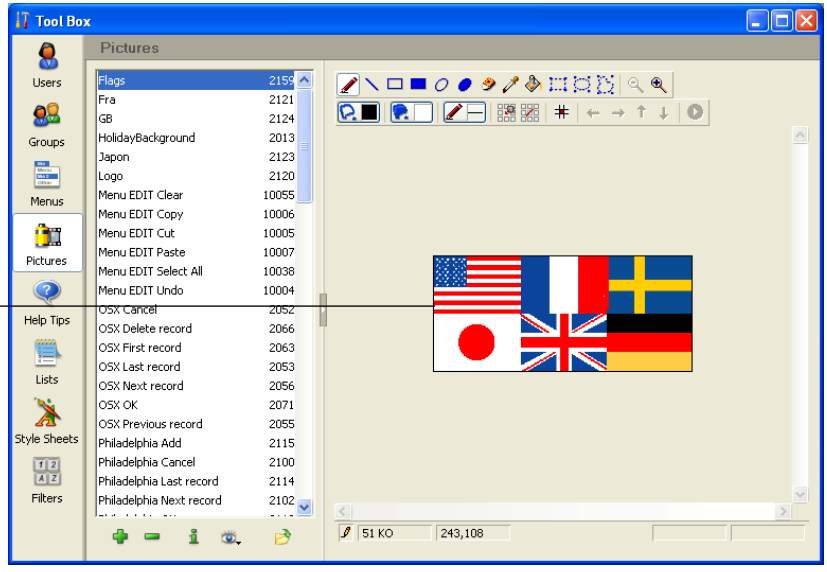


- As a picture button letting the user choose among several choices. In this case, a picture button can be used in place of a pop-up picture menu. With a pop-up picture menu, all choices are displayed simultaneously (as the items in the pop-up menu), while the picture button displays the choices consecutively (as the user clicks the button).
Here is an example of a picture button. Suppose you want to give the users of a custom application the opportunity to choose the interface language for the application. You implement the option as a picture button in a custom properties dialog box, as shown below.



You implement the picture button in the following manner. First, you prepare *one* graphic in which the series of pictures is arranged in a row, a column, or a row-by-column grid.
You can add this graphic to the Picture Library, to a picture variable, to a picture file or to a PICT resource (under Mac OS).

The following illustration shows the graphic in the Picture Library.



The picture button is designed to display only one picture in the series at a time

You can organize pictures as columns, rows, or a row-by-column grid (as shown above). When organizing pictures as a grid, they are numbered from left to right, row by row, beginning with 0.
For example, the second picture of the second row of a grid that consists of four rows and three columns, is numbered 4.

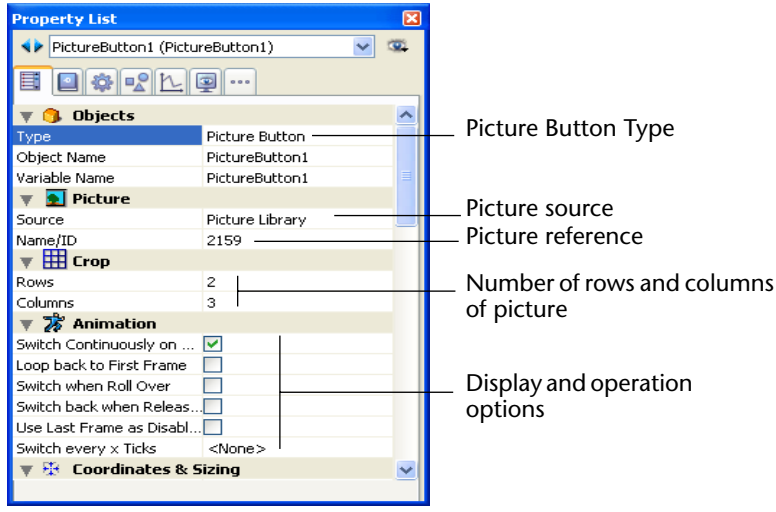
Note The Picture Library includes features that allow you to organize a picture as a table of thumbnails. It also allows you to preview the effects of the current settings.

When a picture has been defined as a table of thumbnails, you can create a picture button by dragging the picture name from the library. For more information on the Picture Library, refer to chapter 14 “Picture Library” on page 891.

Picture buttons have the following specific properties:

- In the “Picture” theme, set the picture source using the “Source” drop-down list. You can choose between Variable, Picture Library, Resource File or File. Once the picture source has been set, enter the name (if the picture is a variable) or the number (if the picture comes from a picture library or resource file) or the pathname (if the picture comes from a picture file) of the picture in the “Name/ID” area; in this latter case, the pathname must be relative to the Resources folder of the database (see the “Automatic Referencing of Picture Files” paragraph on page 456).

- In the “Crop” theme, you can set the number of rows and columns making up the thumbnail table. In our example, we use a picture comprising three columns and two rows.
- In the “Animation” theme, you can set the picture button display mode and operation. You can combine several options. This is described in more detail in the following paragraph.



Note The associated variable of the picture button returns the index number, in the thumbnail table, of the current picture displayed. The numbering of pictures in the table begins with 0.

Display Modes and Picture Button Operation

Here are the different display and operation modes you can set for picture buttons. Naturally, these options can be combined.

- **<No option checked>**
  Displays the next picture in the series when the user clicks; it displays the previous picture in the series when the user holds down the **Shift** key and clicks. When the user reaches the last picture in the series, the picture does not change when the user clicks again. In other words, it does *not* cycle back to the first picture in the series.
- **Switch Continuously**
  Is similar to previous except that the user can hold down the mouse button to display the pictures continuously (i.e., as an animation). When the user reaches the last picture, the object does *not* cycle back to the first picture.

- **Loop back to First Frame**
  Is similar to previous except that the pictures are displayed in a continuous loop. When the user reaches the last picture and clicks again, the first picture appears, and so forth.

- **Switch when Roll Over**
  The contents of the picture button are modified when the mouse cursor passes over it. The initial picture is re-established when the cursor leaves the button's area. This mode is frequently used in multimedia applications or in HTML documents.
  The picture that is then displayed is the last picture of the thumbnail table, unless the Use Last Frame as Disabled option is selected (128). If this option is selected, it is the next-to-last thumbnail that is displayed.

- **Switch back when Released**
  This mode operates with two pictures. It displays the first picture all the time except when the user clicks the button. In that case, the second picture is displayed until the mouse button is released, whereupon it switches back to the first picture. This mode allows you to create an action button with a different picture for each state (idle and clicked). You can use this mode to create a 3D effect or display any picture that depicts the action of the button.

- **Use Last Frame as Disabled**
  This mode allows you to set the last thumbnail as the one to display when the button is disabled. When this mode is selected, 4D displays the last "part" of the referenced picture when the picture button is disabled.
  The thumbnail used when the button is disabled is processed separately by 4D: when you combine this option with the "Switch Continuously" and "Loop Back to First Frame" options, the last picture is excluded from the sequence associated with the button and only appears when it is disabled.

- **Switch every x Ticks**
  This mode allows you to cycle through the contents of the picture button at the specified speed (in ticks). For example, if you pass 10, the thumbnails will change every 10 ticks. In this mode, all other options are ignored.

Note Note that the **Transparent** option ("Appearance" theme) can also be used to set the depiction of the picture button (makes the picture background transparent).

For example, you want to set a button that accepts the following modes: Switch back when Released, Switch when Roll Over and Use Last Frame as Disabled.

In the case of a table of thumbnails that has one row of four columns, each thumbnail corresponds to the following states: Default, Clicked, Roll over and Disabled.

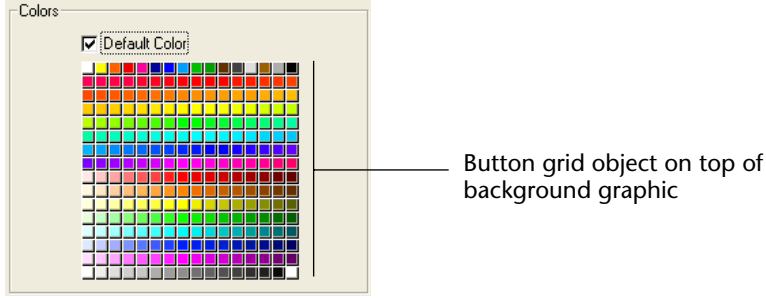
In the Property List you would set the following properties: 1 row, 4 columns as well as the options **Switch back when Released**, **Switch when Roll Over** and **Use Last Frame as Disabled**.

Default Clicked Roll over Disabled

Button Grids

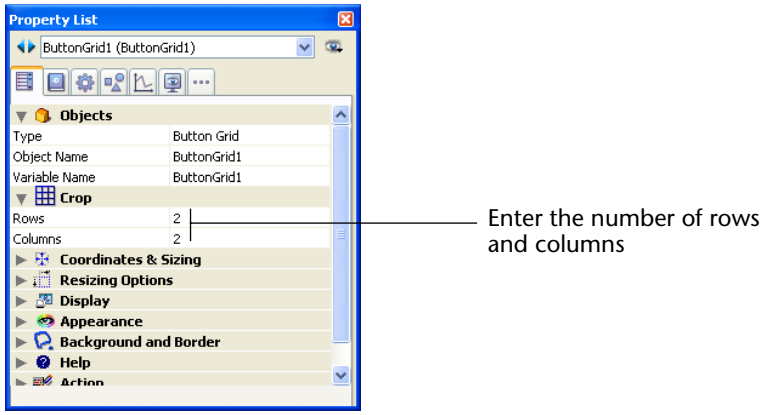
A button grid is a transparent object that is placed on top of a graphic. The graphic should depict a row-by-column array. You can use a button grid object to determine where the user clicks on the graphic. Your object method would use the On Clicked event and take appropriate action depending on the location of the click.

In 4D, a button grid is used as a color palette:



The buttons on the grid are numbered from top left to bottom right. In this example, the grid is 16 columns across by 16 rows down. The button in the top-left position returns 1 when clicked. If the red button at the far right of the second row is selected, the button grid returns 32.

To create the button grid, add a background graphic to the form and place a button grid on top of it. Specify the number of rows and columns in the corresponding entry areas of the “Crop” theme.



Goto Page Action

You can assign the Goto Page action to a button grid. When that action is selected, 4D will automatically display the page of the form that corresponds to the number of the button that is selected in the button grid.

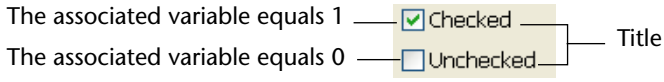
For example, if the user selects the tenth button of the grid, 4D will display the tenth page of the current form (if it exists).
If you want to manage the effect of the selection of a button yourself, select **No action**.

Check Boxes

A check box is used to enter or display binary (true-false) data. It is a type of button. A check box is either selected or deselected. Its effect is controlled by a method. Like all buttons, a check box is set to 0 when the form is first opened. The method associated with a check box executes when the check box is selected.

A check box displays text next to a small square. This text is set in the Title area of the “Objects” theme in the Property List. You can enter a title in the form of an XLIFF or STR# references in this area (see Appendix C on page 1293).

When the user clicks the object, the box is checked. When a check box is checked, it has the value 1. When it is not checked, it has the value 0. Any or all check boxes in a form can be checked or unchecked.



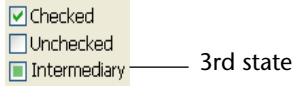
A group of check boxes allows the user to select several options.

Unlike a Boolean field that is formatted as a check box, the values of the check box variable are not stored automatically. You use a method to manage the variable.

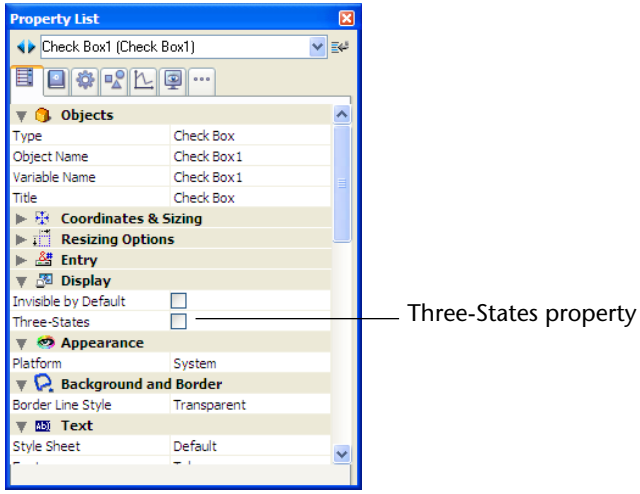
Note 3D check boxes have the same behavior as check boxes but their appearance (for example, the depiction of the checked/unchecked state) is set by the properties of the 3D button family. For more information, refer to the "3D Buttons, 3D Check Boxes and 3D Radio Buttons" paragraph on page 558.

Three-States Check Box

Check box objects accept a third state. This third state is an intermediate status, which is generally used for display purposes. It allows, for example, indicating that a property is present in a selection of objects, but not in each object of the selection.



In order for a check box to take control of this third state, you must set the **Three-States** property in the "Display" theme of the Property List:



This property is only available for standard check boxes associated with numeric variables — 3D check boxes and check boxes for Boolean fields cannot use the Three-States property (a Boolean field cannot be in an intermediary state).

The variable associated with the check box returns the value 2 when the check box is in the third state.

Tip: In entry mode, the Three-States check boxes display each state sequentially, in the following order: unchecked / checked / intermediary / unchecked, etc. The intermediary state is generally not very useful in entry mode; in the code, simply force the value of the variable to 0 when it takes the value of 2 in order to pass directly from the checked state to the unchecked state.

Radio Buttons and Picture Radio Buttons

Radio buttons and picture radio buttons are objects that allow the user to select one of a group of buttons or pictures. A radio button shows a small bull's-eye and text. Picture radio buttons display an icon or picture. They are placed on top of a graphic.

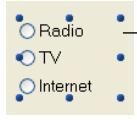
Note 3D radio buttons have identical operation but their appearance is set by the properties of the 3D button family. For more information, refer to the "3D Buttons, 3D Check Boxes and 3D Radio Buttons" paragraph on page 558.

Each type of radio button is selected the same way — you click the object to select it. You can also click a selected picture radio button to deselect it, but you cannot do this with a radio button.

A picture radio button is similar to a highlight button in that it is transparent until selected. When selected, it highlights the picture behind it until another radio button is selected.

The remainder of this section uses the term “radio button” to mean any type of radio button.

Radio buttons are used in coordinated sets: only one button at a time can be selected in the set. To operate in a coordinated manner, a set of radio buttons must be **grouped** in the Form editor. To do this, you can use the **Group** command in the **Object** menu or the corresponding button of the form toolbar.



Coordinated group of standard radio buttons

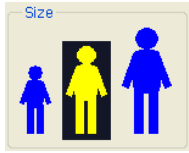
Note In previous versions of 4D, the coordinated behavior of a set of radio buttons was obtained by giving the same first letter to their associated variables (for example, *m_button1*, *m_button2*, *m_button3*, etc.). For compatibility reasons, this principle is kept by default in converted databases. However, you can force the use of this new mode of operation in the Preferences (see the “Compatibility Page” paragraph on page 173).

The effects of radio buttons are controlled with methods. Like all buttons, a radio button is set to 0 when the form is first opened. A method associated with a radio button executes when it is selected.

The following is an example of a group of 3D radio buttons used in a video collection database to enter the speed of the recording (SP, LP, or EP).

Selecting one radio button in a group sets that button to 1 and all the others in the group to 0. Only one radio button can be selected at a time.

The following is an example of a picture radio button. The selected picture radio button appears with a black background:



The value contained in a radio button object is not saved automatically; radio button values must be stored in their variables and managed using methods. Furthermore, the associated variable of a radio button is 0 or 1; a Boolean field stores the True or False value.

Pop-up Menus, Drop-down Lists, and Scrollable Areas

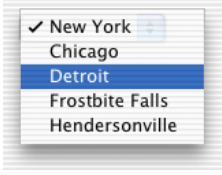
Pop-up menus, drop-down lists, and scrollable areas are objects that allow the user to select from a list. You manage the items displayed in the pop-up menu or scrollable area using an array.

An array is a list of values in memory that is referenced by the name of the array.

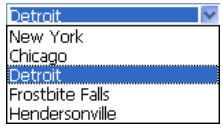
A pop-up menu/drop-down list displays an array as a list of values when you click on it.

The names “Pop-up menu” and “Drop-down list” refer to the same objects; “Pop-up menu” is part of Mac OS terminology and “Drop-down list” is part of Windows. As the following example shows, the appearance of these objects differs slightly according to the platform:

Mac OS



Windows



A scrollable area displays the array in a list that can be scrolled and used to select an item. Here is an example of a scrollable area:

You initialize the object by loading a list of values into an array. You can do this in several ways:

- Enter a list of default values in the object properties. To do this, click on the **Edit...** button in the "Data Source" theme of the Property List. For more information, see "Default Lists of Values" on page 509. The default values are loaded into an array automatically. You can refer to the array using the name of the variable associated with the object.
- Before the object is displayed, execute code that assigns values to the array elements. For example:

```
ARRAY TEXT (aCities;6)
aCities{1}:="Philadelphia"
aCities{2}:="Pittsburg"
aCities{3}:="Grand Blanc"
aCities{4}:="Bad Axe"
aCities{5}:="Frostbite Falls"
aCities{6}:="Green Bay"
```

  In this case also, the name of the variable associated with the object in the form must be *aCities*.
  This code could be placed in the form method and be executed when the On Load form event runs.

- Before the object is displayed, load the values of a list into the array using the LIST TO ARRAY command. For example,

```
LIST TO ARRAY ("Cities";aCities)
```

  In this case, the name of the variable associated with the object in the form must be *aCities*.
  This code would be run in place of the assignment statements shown above.

  If you need to save the user's choice into a field, you would use an assignment statement that runs after the record is accepted. A complete Case statement in the object method might look like this:

```
Case of
  :(Form event=On Load)
    LIST TO ARRAY ("Cities";aCities)
    If (Record number ([People])<0)  `new record
      aCities:=3  `display a default value
    Else   `existing record, display stored value
      aCities:=Find in array (aCities;City)
    End if
```

```
   : (Form event=On Clicked)  `user modified selection
        City:=aCities {aCities}  `field gets new value
   :(Form event=On Validate)
      City:=aCities {aCities}
   :(Form event=On Unload)
      CLEAR VARIABLE (aCities)
End Case
```

In the Events section of the Property List window, you would select each event that you test for in your Case statement.

Arrays always contain a finite number of items. The list of items is dynamic and can be changed by a method. Items in an array can be modified, sorted, and added to.

For information about creating and using an array, refer to the chapter on arrays in the *4th Dimension Language Reference* manual.

Goto Page Action

You can assign the Goto Page standard action to a pop-up menu/drop-down list or a scrollable area ("Action" theme of the Property List). When this action is selected, 4D will automatically display the page of the form that corresponds to the number of the object that is selected in the drop-down list or scrollable area.

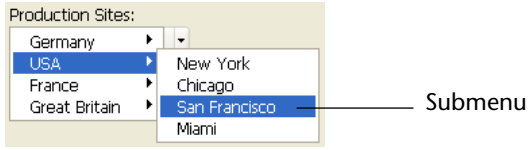
For example, if the user selects the third item of the list, 4D will display the third page of the current form (if it exists). If you want to manage the effect of the selection of an item yourself, select **No action**.

Combo Boxes

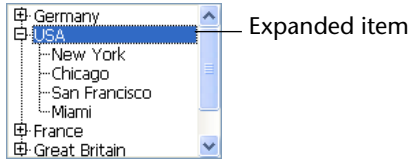
A combo box is similar to a drop-down list, except that the object accepts text entered from the keyboard. You initialize a combo box in exactly the same way as a drop-down list. If the user enters text into the combo box, it fills the 0th element of the array. In other respects, you treat a combo box as an enterable area that uses its array as a set of default values. Use the On Data Change event to manage entries into the enterable area, as you would for any enterable area object.

Hierarchical Pop-up Menus and Hierarchical Lists

A hierarchical pop-up menu has a submenu associated with each item in the menu. Here is an example of a hierarchical menu:



Similarly, a hierarchical list has a sublist associated with each item in the list. Here is an example of a hierarchical list:



You can expand or collapse the hierarchical list by clicking on the plus or minus sign (Windows) or arrows (Mac OS).

You can control whether an item in a hierarchical list can be modified by the user. If an item in a hierarchical list is modifiable, the user can edit it by using the **Alt+click** (Windows) / **Option+click** (under Mac OS) shortcut, or by carrying out a **long click** on the text of the item.
If you populate a hierarchical list using a list created in the Lists editor, you control whether an item in a hierarchical list is modifiable using the **Modifiable Element** option in the Lists editor. For more information, see the "Modifiable Element Option" paragraph on page 920.

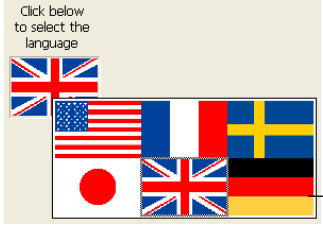
You manage hierarchical pop-up menus and hierarchical lists using the Hierarchical list commands in the language. The principle consists in assigning the hierarchical list reference to the variable associated with the object in the Form editor. For more information, see the "Hierarchical Lists" section of the *4D Language Reference* manual.

Picture Pop-up Menus

A picture pop-up menu is a pop-up menu that displays a two-dimensional array of pictures. A picture pop-up menu can be used to replace a picture button. The creation of the picture to use with a picture pop-up menu is similar to the creation of a picture for a picture button. The concept is the same as a button grid, except that the graphic is used as a pop-up menu instead of a form object.

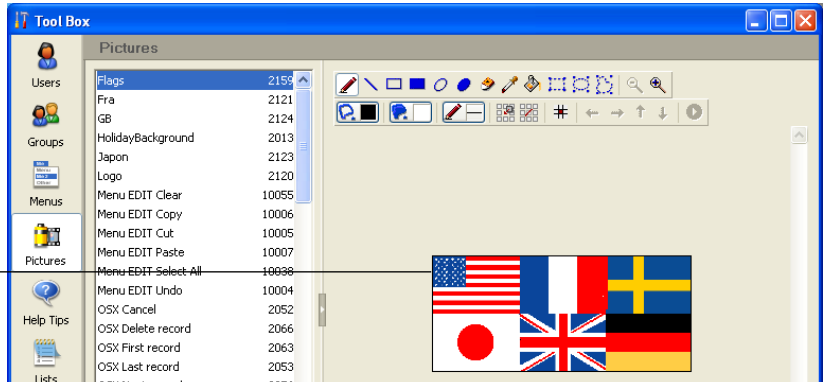
To create a picture pop-up menu, you need to refer to a picture. The following example uses the picture that was defined for picture buttons. In this case, it allows you to select the interface language by selecting it from a picture pop-up menu.

Each language is represented by the corresponding flag.



Picture pop-up menu

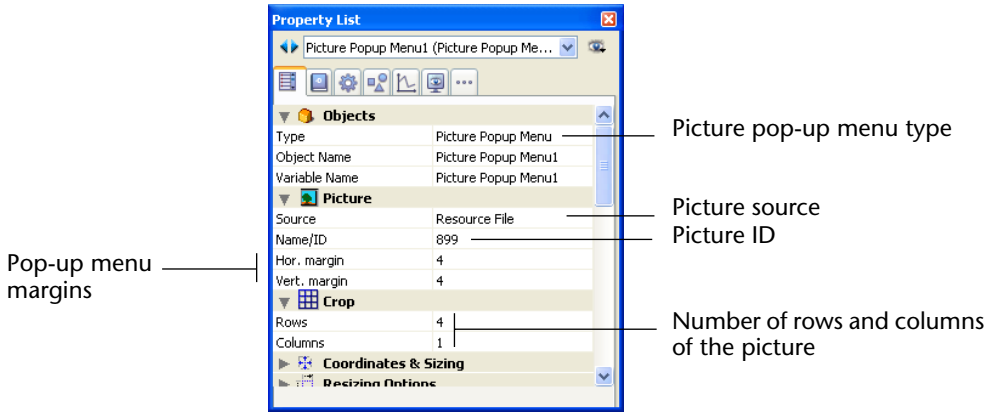
As with a picture button, a picture pop-up menu uses a picture that is organized in columns and rows. You can place that picture in the Picture Library, in a picture variable or in a picture file.



The picture pop-up menu object displays the whole picture

Note The Picture Library includes features that allow you to organize a picture as a table of thumbnails. It also allows you to preview the effects of the current settings. When a picture is defined as a table of thumbnails, you can create a picture pop-up menu by dragging the picture name into the form while pressing the Shift key.
For more information on the Picture Library, refer to chapter 14 “Picture Library” on page 891.

Various specific properties can be used to configure picture pop-up menus:



- In the “Picture” theme, set the picture source using the “Source” drop-down list. You can choose between Variable, Picture Library, Resource File or File. Once the picture source has been set, enter the name (if the picture is a variable) or number (if the picture comes from a picture library or a resource file) or pathname (if the picture comes from a picture file) of the picture in the “Name/ID” area; in this latter case, the pathname must be relative to the Resources folder of the database (see the “Automatic Referencing of Picture Files” paragraph on page 456).
- In the “Crop” theme, set the number of rows and columns making up the thumbnail table.
- The *Hor. margin* and *Vert. margin* options create a margin between the edge of the menu and the picture. Enter values expressed in pixels.

Operation of Picture Pop-up Menu

It is not possible to assign a standard action to a picture pop-up menu. Picture pop-up menus are managed using methods.

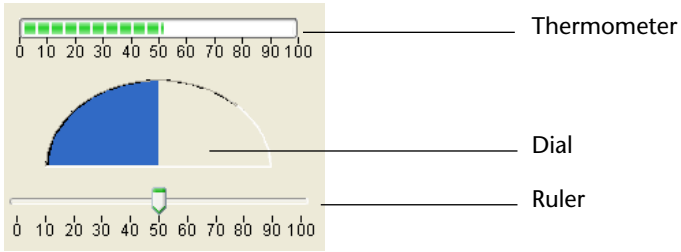
As with button grids, variables associated with picture pop-up menus are set to the value of the selected element in the picture pop-up menu. If no element is selected, the value is 0. Elements are numbered, row by row, from left to right starting with the top row.

Indicators

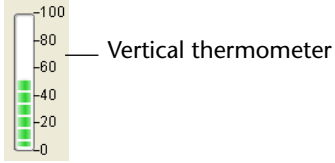
Thermometers, rulers, and dials are objects that display a value graphically. The three objects work in the same way; they differ only in their appearance. We refer to these three objects as *indicators*.

You can use indicators either to display or set values. For example, if a thermometer is given a value by a method, it displays the value. If the user drags the indicator point, the value changes.

The value can be used in another object such as a field or an enterable or non-enterable object.

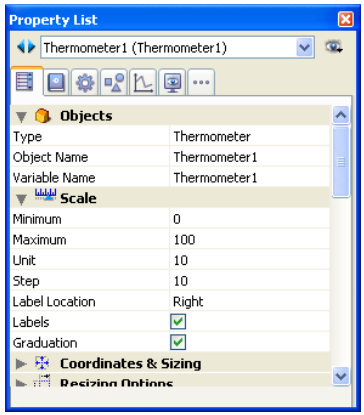


You can display horizontal or vertical indicators (except for dials). The type of indicator is determined by the shape of the object that you draw:



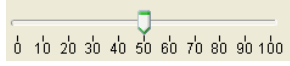
In addition to the standard positioning and appearance settings, you can set some other specific properties for indicators: minimum value, maximum value, units for the tick marks, the minimum steps permitted by the indicator as well as display options. You can also set the display format of an indicator's label (for more information on display formats, refer to "Display Formats" on page 521).

These specific properties are set in the “Scale” theme of the Property List:

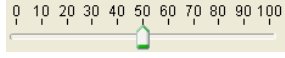


Options related to labels and scale are only applied to Ruler and Thermometer objects. Here is a description of each property:

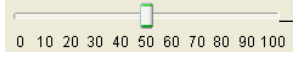
- **Minimum** and **Maximum**: Minimum and maximum values of the indicator.
- **Unit**: Scale display unit.
- **Step**: Minimum interval accepted between values during use.
- **Label Location**: Position of labels when they are displayed.
  - **Bottom**: Displays labels to the right of or below the indicator.

  - **Top**: Displays labels to the left of or above the indicator.

- **Labels**: Displays/Hides labels.
- **Graduation**: Displays/Hides the graduations next to the labels.

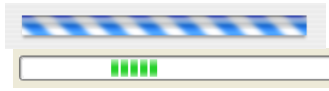
Execute Object Method

There is an additional property in the “Action” theme of the Property List: **Execute Object Method**.

When this option is checked, the object method will be executed with the *On Data Change* event when the user changes the value of the indicator. By default, the method is executed after the modification.

"Barber Shop" Thermometer

You can use "Barber shop" type parameters in forms. This type of thermometer displays a continuous animation. "Barber shop" thermometers are generally used to indicate to the user that the program is in the process of carrying out a long operation.



Barber shop under Mac OS

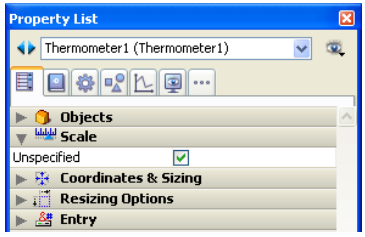
Barber shop under Windows XP

▶ To add a "Barber shop" to a form:

1 **Create a "Thermometer."**

2 **Check "Unspecified" for the Scale properties in the Property List.**

Activating this property will hide the other scale options for the thermometer.



When the form is executed, the thermometer is not animated. The animation is handled by passing a value to the variable associated with the thermometer:

- 1 = Start animation,
- 0 = Stop animation.

Note "Barber shop" thermometers only work when the appearance is set to System or Printing.

Programmed Management of Indicators

The variable associated with the indicator controls the display. You place values into, or use values from, the indicator using methods. For example, a method for a field or enterable object could be used to control a thermometer.

The method:

```
vTherm:=[Employees]Salary
```

assigns the value of the Salary field to the vTherm variable. This method would be attached to the Salary field.

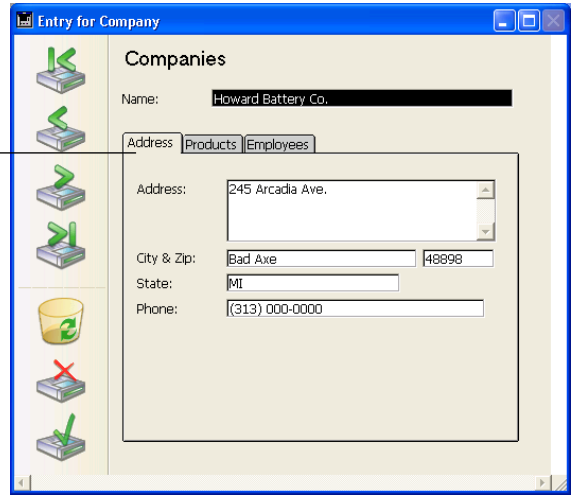
Conversely, you could use the indicator to control the value in a field. The user drags the indicator to set the value. The method:

[Employees]Salary:=vTherm

assigns the value of the thermometer to the Salary field. As the user drags the indicator, the value in the Salary field changes.

Tab Controls

A tab control creates an object that lets the user choose among a set of virtual screens that are enclosed by the tab control object. Each screen is accessed by clicking its tab. The following multi-page form uses a tab control object:



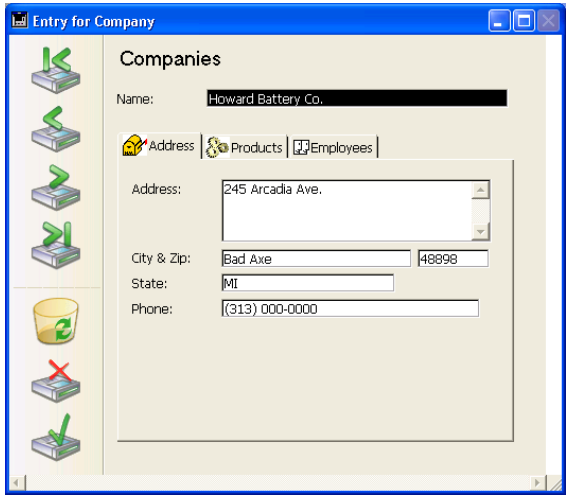
To navigate from screen to screen, the user simply clicks the desired tab.

The screens can represent pages in a multi-page form or an object that changes when the user clicks a tab. If the tab control is used as a page navigation tool, then the GOTO PAGE command or the **Goto Page** standard action would be used when a user clicks a tab.

Another use of the tab control is to control the data that is displayed in a subform or grouped scrollable areas. For example, a Rolodex could be implemented using a tab control. The tabs would display the letters of the alphabet and the tab control's action would be to load the data corresponding to the letter that the user clicked.

Each tab can display labels or labels and a small icon. If you include icons, they appear to the left of each label.

Here is an example of a tab control that uses icons:



When you create a tab control, 4D manages the spacing and placement of the tabs. You only need to supply the labels in the form of an array, or the icons and labels in the form of a hierarchical list.

If the tab control is wide enough to display all the tabs with both the labels and icons, it displays both.

If the tab control is not wide enough to display both the labels and icons, 4D displays the icons only. If it can't fit all the icons, it places scroll arrows to the right of the last visible tab. The scroll arrows allow the user to scroll the icons to the left or right.

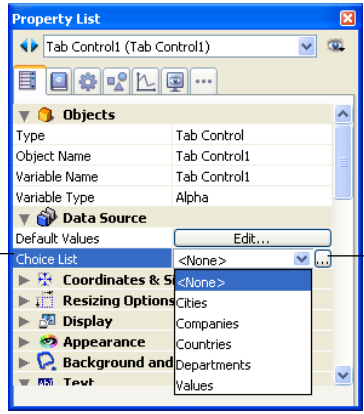
Under Mac OS, in addition to the standard position (top), the tab controls can also be aligned to the left, to the right, or below (see the "Modifying Direction (Mac OS Only)" paragraph on page 588).

Adding Labels to a Tab Control

There are several ways to supply the labels for a tab control.

- You can associate the tab control with a list of default values, which can be accessed using the **Edit** button next to the Default Values property found in the "Data Source" theme of the Property List. For more information about this, refer to the "Default Lists of Values" paragraph on page 509. Default values are automatically loaded into an array. You can refer to this array using the name of the variable associated with the tab control.

- You can create a list using the Lists editor and assign the list to the tab control as a choice list, as shown below



If you like, you can associate a small icon with each list item using the Lists editor. For more information about this, refer to the "Adding a Small Icon to an Item" paragraph on page 913.

- You can create a Text array that contains the names of each page of the form. This code must be executed before the form is presented to the user. For example, you could place the code in the object method of the tab control and execute it when the On Load event occurs.

```
ARRAY TEXT (arrPages;3)
arrPages {1}:="Name"
arrPages {2}:="Address"
arrPages {3}:="Notes"
```

Note You can also store the names of the pages in a hierarchical list and use the Load list command to load the values into the array.

GOTO PAGE Command

Use the GOTO PAGE command in the tab control's method:

```
GOTO PAGE (arrPages)
```

The command is executed when the On Clicked event occurs.

You should then clear the array when the On Unload event occurs.

Here is an example object method:

```
Case of
  :(Form event=On Load)
     LIST TO ARRAY ("Tab Labels";arrPages)
  :(Form event=On Clicked)
     GOTO PAGE (arrPages)
  :(Form event=On Unload)
     Clear variable (arrPages)
End Case
```

Goto Page Action

You can assign the Goto Page action to a tab control. When this action is selected, 4D will automatically display the page of the form that corresponds to the number of the tab control that is clicked.

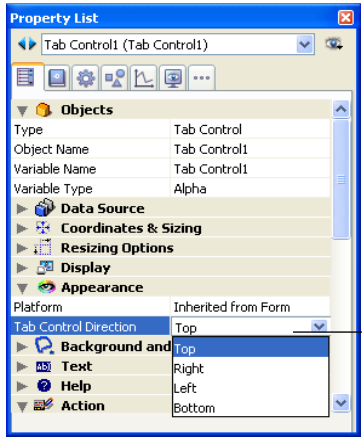
For example, if the user clicks the third tab control, 4D will display the third page of the current form (if it exists). If you want to manage the tab control programmatically, select **No action**.

Modifying Direction (Mac OS Only)

You can set the direction of tab controls in your forms. This property is available on all the platforms but can only be displayed under Mac OS, when the platform interface is "System." You can choose to place the tab controls on **top** (standard), to the **left**, to the **right**, or on the **bottom**.

Tabs to the left

The tab control direction is set in the “Appearance” theme of the Property List:



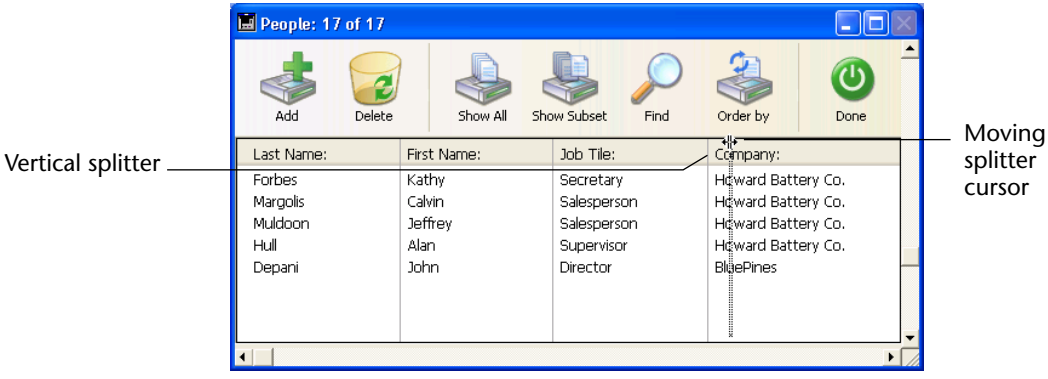
When tab controls with a custom direction are displayed under Windows or with the “Printing” platform appearance, they automatically return to the standard direction (top).

Splitters

A splitter divides a form into two areas, allowing the user to enlarge and reduce the areas by moving the splitter one way or the other. A splitter can be either horizontal or vertical.

The splitter takes into account each object’s resizing properties, which means that you can completely customize your database’s interface. A splitter may or may not be a “pusher.”

The splitter is generally used in output forms so that columns can be resized:



Here are some of the splitter's general characteristics:

- You can place as many splitters as you want in any type of form and use a mixture of horizontal and vertical splitters in the same form.
- A splitter can cross (overlap) an object. This object will be resized when the splitter is moved.
- Splitter stops are calculated so that the objects moved remain entirely visible in the form or do not pass under/next to another splitter. When the Pusher property is associated with a splitter, its movement to the right or downward does not encounter any stops.
- If you resize a form using a splitter, the new dimensions of the form are saved only while the form is being displayed. Once a form is closed, the initial dimensions are restored.

Once it is inserted, the splitter appears as a line. You can modify its border style to obtain a thinner line or change its color. You can also use an invisible button with the Automatic Splitter standard action as a splitter.

Interaction with the Properties of Neighboring Objects

In a form, splitters interact with the objects that are around it according to these objects' resizing options:

Resizing options for the object(s)	Object(s) above the horizontal splitter or to the left of the vertical splitter[1]	Object(s) below the horizontal splitter or to the right of the vertical splitter	
		non-"Pusher" splitter	"Pusher" splitter
None	Remain as is	Are moved with the splitter (position relative to the splitter is not modified) until the next stop. The stop when moving to the bottom or right is either the window's border, or another splitter	Are moved with the splitter (position relative to the splitter is not modified) indefinitely. No stop is applied (see the next paragraph)
Resize	Keep original position(s), but are resized according to the splitter's new position		
Move	Are moved with the splitter		

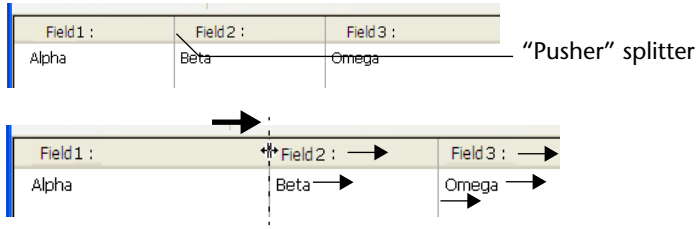
1. You cannot drag the splitter past the right (horizontal) or bottom (vertical) side of an object located in this position.

Note An object completely contained in the rectangle that defines the splitter is moved at the same time as the splitter.

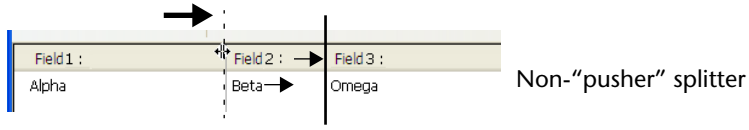
Pusher Property

The **Pusher** property is available for splitter objects in the "Resizing Options" theme of the Property List. When a splitter object has this property, other objects to its right (vertical splitter) or below it (horizontal splitter) are pushed at the same time as the splitter, with no stop.

Here is the result of a "pusher" splitter being moved:



When this property is not applied to the splitter, the result is as follows:



This property is checked by default for new databases and disabled in the forms of converted databases.

Managing Splitters Programmatically

You can associate an object method with a splitter and it will be called with the On Clicked event throughout the entire movement.

A variable of the Longint type is associated with each splitter. This variable can be used in your object and/or form methods. Its value indicates the splitter's current position, in pixels, in relation to its initial position.

- If the value is negative: the splitter was moved toward the top or toward the left,
- If the value is positive: the splitter was moved toward the bottom or toward the right,
- If the value is 0: the splitter was moved to its original position.

You can also move the splitter programmatically: you just have to set the value of the associated variable. For example, if a vertical splitter is associated with a variable named *split1*, and if you execute the following statement: split1:=-10, the splitter will be moved 10 pixels to the left — as if the user did it manually.

The move is actually performed at the end of the execution of the form or object method containing the statement.

The Automatic Splitter action allows you to create custom splitters in your forms. You can assign this action to an object of the **invisible button** type. When an invisible button is assigned this standard action, it acts exactly as a splitter. If, for example, you paste a picture on the invisible button, you can create a custom interface for your splitters. For more information about this type of button, refer to "Buttons" on page 551.

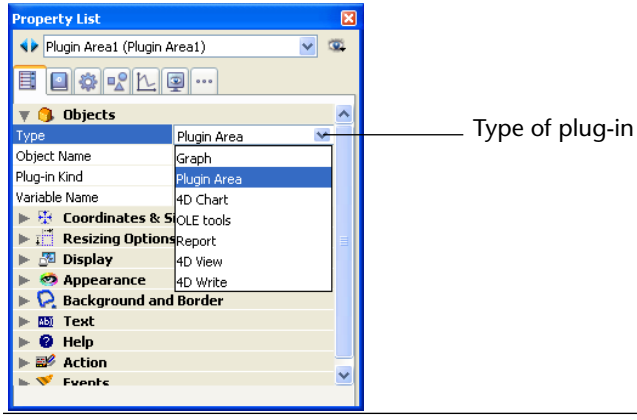
Plug-in Areas

A *plug-in area* is an area on the form that is completely controlled by a 4D plug-in.

When opening a database, 4D creates an internal list of the plug-ins installed in your database. Once you have inserted a Plug-in Area in a form, 4D lists the available plug-ins in the Property List window. By default, 4D Chart, OLE tools (Windows only) and Report are available.

Note Some plug-ins cannot be used in forms or external windows. When a plug-in cannot be used in a form, it does not appear in the plug-in list of the Property List.

To specify the plug-in to be used in the area, select its name directly in the Type list ("Objects" theme):



Note When the object type chosen is Plugin Area, an **Advanced** button may be enabled if advanced options are provided by the author of the plug-in. In this case, you can click this button to set these options. Because the Advanced options dialog box is under the control of the author of the plug-in, information about these Advanced options is the responsibility of the distributor of the plug-in.

If you draw a plug-in area that is too small, 4D will display it as a button whose title is the name of the variable associated with the area. During execution, the user can click on this button in order to open a specific window displaying the plug-in.

Built-in Plug-in Areas

By default, 4D offers the following types of plug-ins:

- **4D Chart**: A 4D Chart area is used to build and display a chart in a form. This area can be controlled using its own menu bar or by programming. For more information about the 4D Chart user interface, refer to chapter 21 "Graphs" on page 1053. For more information about programming 4D Chart areas, refer to the 4D Chart *Language Reference* manual.
- **OLE tools** (Windows only): Under Windows, an OLE area lets you open a window containing documents coming from other applications within your database. This subject is described in detail in the *OLE_Tools* manual.

Note It is also possible to create an OLE area directly using the **Insert an OLE Object** command in the **Object** menu.

- **Report**: A Report area is used to insert a Quick report into a form. The Quick report area may display data in the form of a table, carry out summary calculations, etc. This area can be controlled using its own menu bar or using the language commands of 4D. For more information, refer to chapter 10 "Quick Reports" on page 699 and the *Language Reference* manual of 4D.

Installing Plug-ins

To install a plug-in in your 4D environment, you first need to quit 4D. Plug-ins are loaded when you launch 4D. For more information about the installation of plug-ins, please refer to the "Installing Plug-ins" paragraph on page 47.

Using Plug-ins

The ability to incorporate plug-ins into forms gives you unlimited possibilities when creating custom applications. A plug-in can perform a simple task such as displaying a digital clock on a form, or a complex task such as providing full-featured word processing, spreadsheet, or graphics capabilities. Many of these capabilities are already available for the 4D environment in the form of 4D plug-ins. For example:

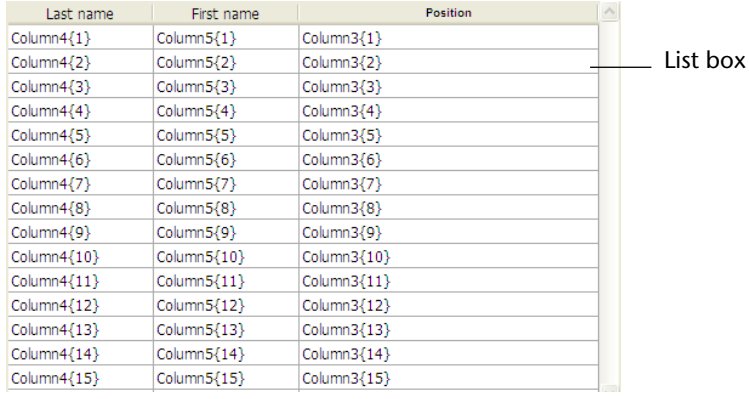
- 4D Write, a word processing application,
- 4D View, an advanced spreadsheet and list management application.

For more information, refer to the documentation that comes with these plug-ins.

If you are interested in designing your own plug-ins, you can receive extensive information about writing and implementing plug-ins. 4D provides a complete kit to help you write custom plug-ins. For more information, contact 4D, Inc. (http://www.4d.com).

List Boxes

List boxes are complex active objects that allow displaying and entering data as synchronized columns. They have the same basic features as "grouped scrollable areas," as well as new expanded possibilities (value entry, column sorting, customized appearance, moving of columns, etc.).
A list box type object can be completely set using the 4D Form editor and can also be handled programmatically.



This paragraph details the principles related to the creation and configuration of list box type objects in the Form editor. For more information on the programmed management of these objects, please refer to the "List Box" chapter of the *4D Language Reference* manual.

Selection Type and Array Type

There are two types of list boxes: **selection type list boxes** and **array type list boxes**.

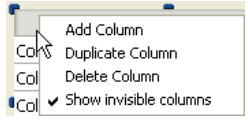
- **Selection type list boxes**: The number of rows is based on the current selection or on a named selection. Each column is associated with a field (for example [Employees]LastName) or a 4D expression. The expression can be based on one or more fields (for example [Employees]FirstName+" "[Employees]LastName) or it may simply be a formula (for example String(Milliseconds)). The expression can also be a project method, a variable or an array item.
  In the case of a list box based on the current selection, any modification done from the database side is automatically reflected in the list box, and vice versa. The current selection is therefore always the same in both places.

- **Array type list boxes**: The number of rows is based on the number of array elements. Each column of the list box is associated with a 4D array. By default, 4D assign the name "ColumnX" to each column variable, and thus to each associated array. You can change it in the column properties. With this type of list box, the values entered or displayed are managed using the 4D language. You can also associate a choice list with a column in order to control data entry (see the "List Box Column Specific Properties" paragraph on page 605).

  It is not possible to combine columns associated with fields (or expressions) and columns associated with arrays in the same list box.

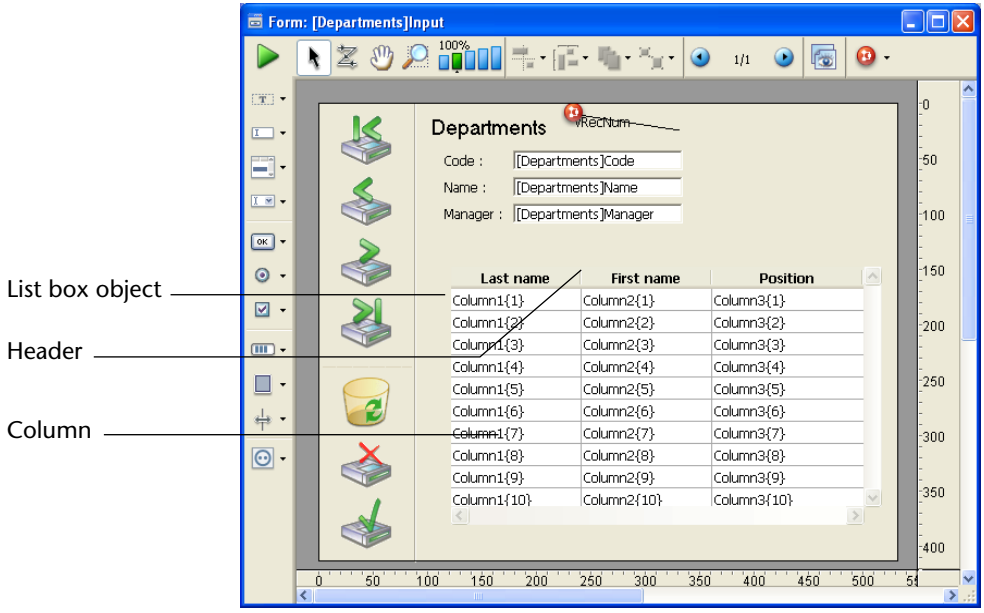
Main Characteristics

A list box contains one or more columns whose contents are automatically synchronized. By default, when you create a list box, it contains a single column. You can modify the number of columns (add, duplicate, or delete a column) using the contextual menu (click on a column or column header) or in the list box properties.



The number of columns is, in theory, unlimited (it depends on the machine resources).

A list box is composed of three distinct parts: the **object** in its entirety, **columns** and column **headers**. In the Form editor, these parts can also be selected separately. Each part has its own object and variable name as well as specific properties:



For example, the number of columns or the alternating color of each row is set in the list box object properties, the width of each column is set in the column properties, and the font of the header is set in the header properties.

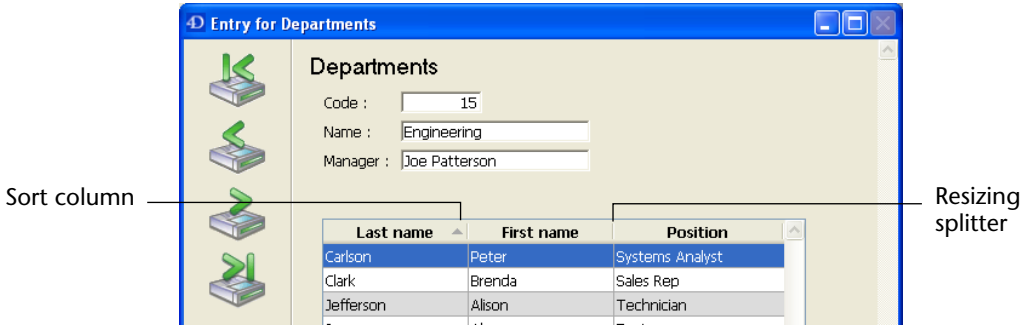
Note In selection type list boxes, columns do not have a variable name.

Using List Boxes

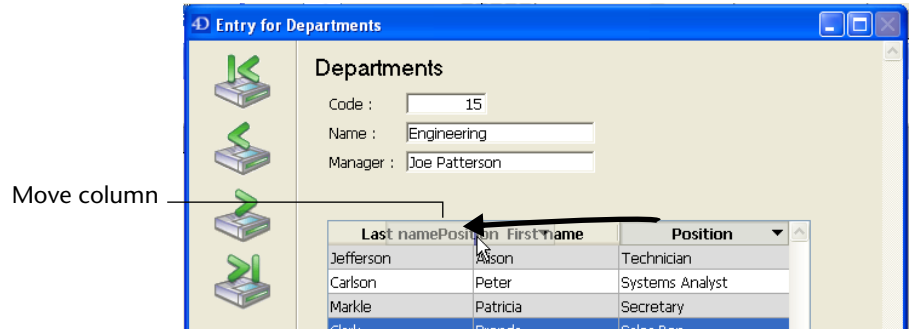
During execution, list boxes allow displaying and entering data as lists. To make a cell editable (if entry is allowed for the column), simply click twice on the value that it contains:

Last name	First name
James	Henry
Jameson	Marc

It is possible to sort column values by clicking on a header (standard sort). The sort is alphanumerical and alternately ascending/descending on multiple clicks. All columns are automatically synchronized.
It is also possible to resize each column:

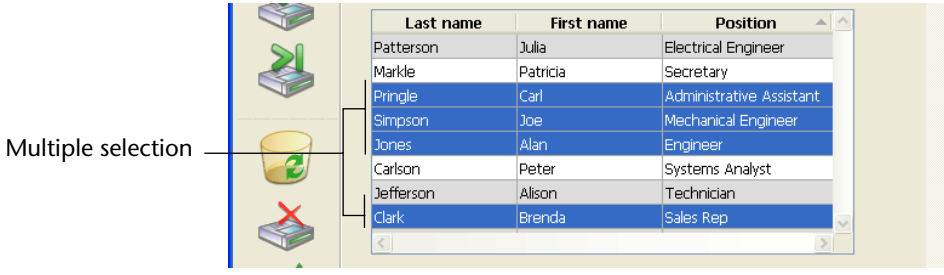


The user can modify the order of columns and rows by moving them using the mouse (if this action is authorized):



Note In selection type list boxes, it is not possible to change the order of the rows by moving them with the mouse.

Finally, the user can select one or more rows using the standard shortcuts: **Shift+click** for an adjacent selection and **Ctrl+click** (Windows) or **Command+click** (Mac OS) for a non-adjacent selection.

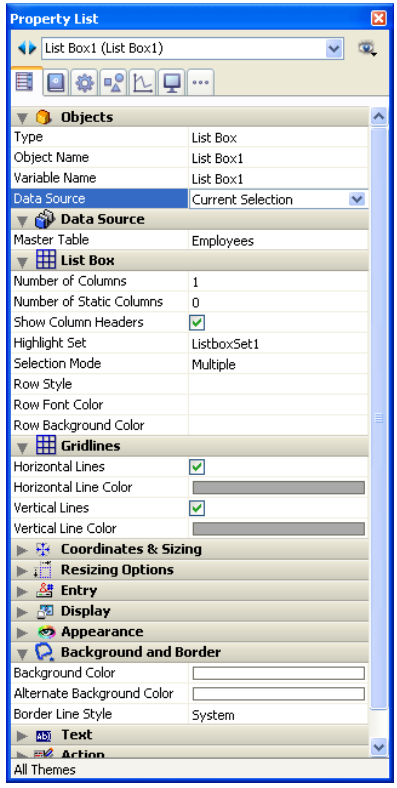


All these characteristics can be handled using the list box, column and header properties. They are detailed in the following paragraphs.

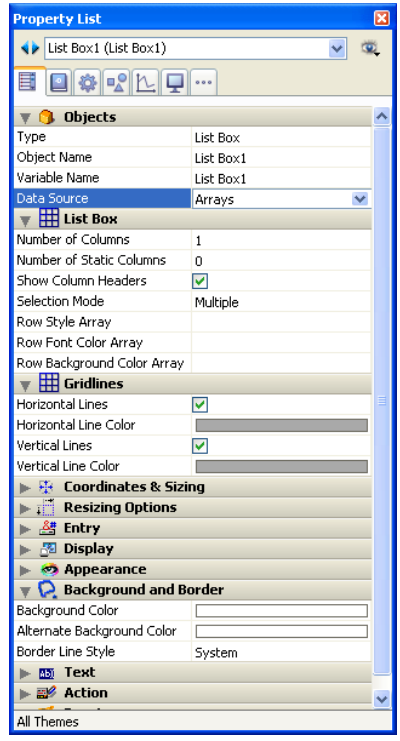
List Box Specific Properties

When you select a List Box object in the Form editor, the Property List displays the following specific properties:

Selection type list box

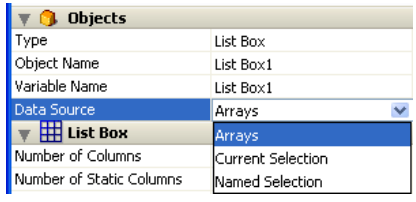


Array type list box



Objects Theme

This theme includes the **Data Source** property that is used to specify the type of list box:



- Select the **Current Selection** option if you want to use expressions, fields or methods whose values will be evaluated for each record of the current selection of a table.

- Select the **Named Selection** option if you want to use expressions, fields or methods whose values will be evaluated for each record of a named selection.
- Select the **Arrays** option if you want to use array elements as the rows of the list box.
  The **Arrays** option is required when you want to be able to retrieve the result of an SQL query in a list box (see the "Displaying the Result of an SQL Query in a List Box" paragraph on page 615).

Data Source Theme

This theme appears for selection type list boxes only. It contains the **Master Table** property for list boxes based on the current selection or the **Named Selection** property for list boxes based on a named selection.

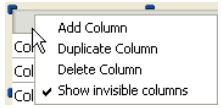
- **Master Table**: Specifies the table whose current selection will be used. This table and it current selection will form the reference for the fields associated with the columns of the list box (field references or expressions containing fields). Even if some columns contain fields from other tables, the number of rows displayed will be defined by the master table.
  The menu associated with this property displays all the database tables, regardless of whether the form is related to a table (table form) or not (project form). By default, the property displays the first table of the database. For more information about the behavior of this property, see the "Display of Fields in List Boxes" paragraph on page 612.
- **Named Selection**: Specifies the named selection to be used. You must enter the name of a valid named selection. It can be a process or interprocess named selection. The contents of the list box will be based on this selection. The named selection chosen must exist and be valid at the time the list box is displayed, otherwise the list box will be displayed blank. If you leave the name area blank, the list box will also be displayed blank.

Note Named selections are ordered lists of records. They are used to keep the order and current record of a selection in memory. For more information, please refer to the *Language Reference* manual of 4D.

List Box Theme

This theme groups together all the basic object properties.

- **Number of Columns**: Number of columns displayed in the list box (1 by default). You can modify the number of columns (add, duplicate or delete a column) using the contextual menu (click on a column or column header):



- **Number of Static Columns**: Number of columns that cannot be moved during execution. This value indicates the number of static columns starting from the first column of the list box. To prevent any column being moved, this value must be equal to the total number of columns. This property takes invisible columns into account.
By default, the property value is 0, which means that all columns can be moved.

- **Show Column Headers**: Allows displaying or hiding column headers (displayed by default).

- **Highlight Set**: This property is added for selection type list boxes only. It is used to specify the set to be used to manage highlighted records in the list box (when the **Arrays** data source is selected, a Boolean array with the same name as the list box is used, see the *Language Reference* manual).
4D creates a default set, named *ListBoxSet1*, that you can modify if necessary (it can be a process or interprocess set). It is then automatically maintained by 4D. If the user selects one or more rows in the list box, the set is immediately updated. If you want to select one or more rows by programming, you can apply the commands of the "Sets" theme to this set.

Notes

• The highlighted status of the list box rows and the highlighted status of the table records are completely independent.
• If the "Highlight Set" property does not contain a name, it will not be possible to make selections in the list box.

- **Selection Mode**: Used to specify the selection mode for rows in the list box. Three modes are available:
  - **None**: No row can be selected and no data can be entered. Selection and data entry can only be managed by programming. Clicking or double-clicking on the list will have no effect (even if the **Enterable** option is checked) but the On Clicked and On Double Clicked events can be generated. With this mode, the developer has full control over selections (using the Highlight Set) and over data entry (using the EDIT ITEM command).
    The On Selection Change and On Before Data Entry events are not generated. On the other hand, the On After Edit event can be generated when data is entered by the user through the EDIT ITEM command.
  - **Single**: Only one row can be selected at a time.
  - **Multiple**: Several rows can be selected (adjacent or not) using standard shortcuts: **Shift+click** for an adjacent selection and **Ctrl+click** (Windows) or **Command+click** (Mac OS) for a non-adjacent selection.
- **Row Style Array** (array type list box) / **Row Style** (selection type list box): Used to apply a custom character style to each row of the list box.
  - For array type list boxes, you must enter the name of a Longint type array. Each element of this array corresponds to a row of the list box; the array must have the same size as the arrays associated with columns. To fill in the array (using a method), use the "Font Styles" theme constants. By adding constants, you can combine styles. To apply the style set in the properties of the list box to a row, pass the value -255 to the element of the corresponding array.
  - For selection type list boxes, you must enter an expression or a variable (except for an array). The expression or variable will be evaluated for each row displayed. You can use the Formula editor to specify an expression. To do this, click on the [...] button which appears when you select the area. You can use the constants of the "Font Styles" theme.
    The following example uses a variable name: enter *CompanyStyle* in the **Row Style** area and, in the form method, write the following code:

```
CompanyStyle:=Choose([Companies]ID;Bold;Plain;Italic;Underline)
```

- **Row Font Color Array** (array type list box) / **Row Font Color** (selection type list box): Used to apply a custom font color to each row of the list box. You must use RGB color values. For more information about this, please refer to the description of the SET RGB COLORS command in the 4D *Language Reference* manual.
  - For array type list boxes, you must enter the name of a Longint type array. Each element of this array corresponds to a row of the list box; the array must have the same size as the arrays associated with columns. You can use the constants of the "SET RGB COLORS" theme. To apply the font color specified in the list box properties to a row, pass the value -255 to the corresponding element of the array.
  - For selection type list boxes, you must enter an expression or a variable (except for an array). The expression or variable will be evaluated for each row displayed. You can use the Formula editor to specify an expression. To do this, click on the [...] button which appears when you select the area. You can use the constants of the "SET RGB COLORS" theme.
    The following example uses a variable name: enter *CompanyColor* in the **Row Font Color** area and, in the form method, write the following code:

```
CompanyColor:=Choose([Companies]ID;Default background color;
                     Default light shadow color;Default foreground color;
                     Default dark shadow color)
```

  - **Row Background Color Array** (array type list box) / **Row Background Color** (selection type list box): Used to apply a custom background color to each row of the list box. You must use RGB color values. For more information about this, please refer to the description of the SET RGB COLORS command in the 4D *Language Reference* manual.
  - For array type list boxes, you must enter the name of a Longint type array. Each element of this array corresponds to a row of the list box; the array must have the same size as the arrays associated with columns. You can use the constants of the "SET RGB COLORS" theme. To apply the background color specified in the list box properties to a row, pass the value -255 to the corresponding element of the array.

- For selection type list boxes, you must enter an expression or a variable (except for an array). The expression or variable will be evaluated for each row displayed. can use the Formula editor to specify an expression. To do this, click on the [...] button which appears when you select the area. You can use the constants of the "SET RGB COLORS" theme.

Gridlines Theme

This theme groups together all of the properties linked to the grid displayed in the list box object.

- **Horizontal Lines**: Displays or hides the horizontal lines of the list box (displayed by default).
- **Horizontal Line Color**: Defines the color of horizontal lines in the list box (gray by default).
- **Vertical Lines**: Displays or hides the vertical lines of the list box (displayed by default).
- **Vertical Line Color**: Defines the color of vertical lines in the list box (gray by default).

Coordinates & Sizing Theme

This theme groups together all the properties related to the coordinates, width and height of the list box.
The **Row Height** property is specific: it allows you to set the height of the list box rows (in pixels). This height is also applied to headers. The row height is set, by default, according to the platform and the font size.

Appearance Theme

This theme groups together all the standard properties related to the platform interface of the list box. You can also specify in this theme whether or not the list box will have horizontal and vertical scroll bars.

Background and Border Theme

This theme groups together properties related to the background color of rows as well as the border style.

- **Background Color**: Allows setting the background color of the list box. This color is used for the entire object with the exception of headers (if displayed).
- **Alternate Background Color**: Allows setting a different background color for odd-numbered rows in the list box. Using an alternating background color makes reading arrays easier.

- **Border Line Style**: Allows setting a standard style for the list box object border.

Text Theme

This theme groups together all the standard properties related to texts displayed in the list box (style sheet, font, attributes, etc.). Note that specific text properties can be set for each column and header.

Action Theme

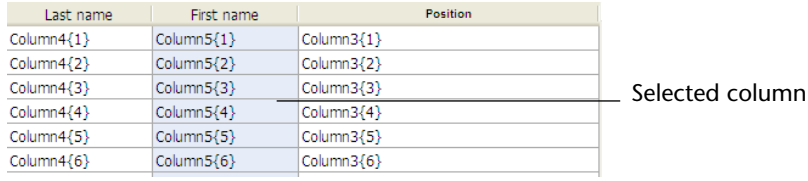
This theme groups together all the properties related to the dynamic behavior of the list box.

- **Method (Edit...)**: This button displays the method of the list box object (note that each column can also contain an object method).
- **Draggable** and **Droppable**: Activates the drag-and-drop functions of the list boxes, which allows a list box row to be dragged and dropped onto another list box or another 4D object and vice versa. Only list box rows are concerned; it is not possible to drag and drop columns (however, it is still possible to move columns inside the same list box). Drag and drop in list boxes is managed using standard 4D drag-and-drop mechanisms (On Drop and On Drag Over form events, DRAG AND DROP PROPERTIES and Drop position commands). For more information, refer to the *Language Reference* manual.
- **Movable Rows** (array type list boxes only): Authorizes the movement of rows during execution. This option is checked by default. It is not available for selection type list boxes.
- **Sortable**: Allows sorting column data by clicking the header. This option is checked by default.
  Picture type arrays (columns) cannot be sorted using this mechanism. In list boxes based on a selection of records, the standard sort function is available only:
  - When the data source is **Current Selection**,
  - With columns associated with fields (of the Alpha, Number, Date, Time or Boolean type).
    In other cases (list boxes based on named selections, columns associated with expressions), the standard sort function is not available.

  A standard list box sort changes the order of the current selection in the database. However, the highlighted records and the current record are not changed.
  A standard sort synchronizes all the columns of the list box, including calculated columns.

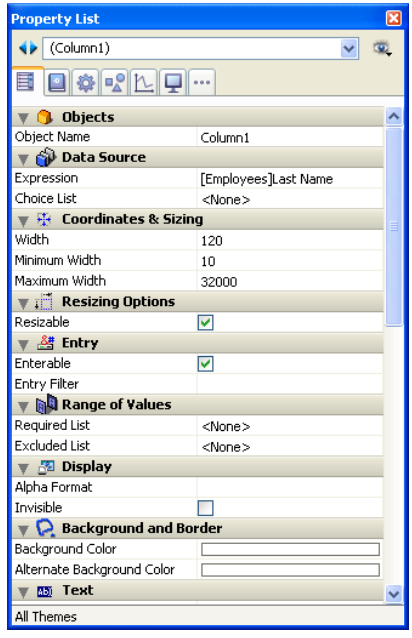
List Box Column Specific Properties

You can select a list box column in the Form editor by clicking on it when the List Box object is selected:

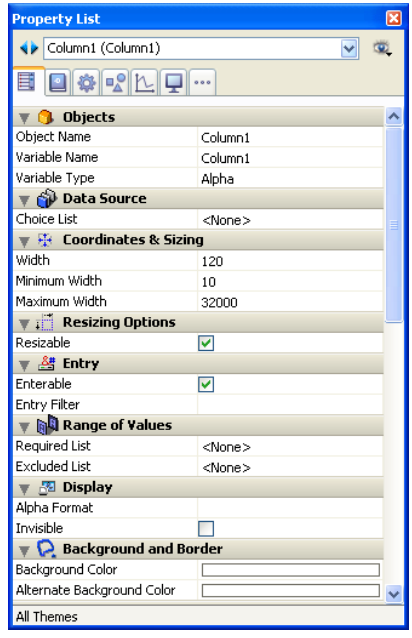


The following specific properties are displayed:

Selection type list box



Array type list box

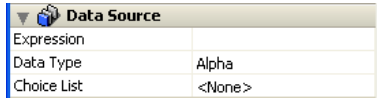


Note The **Variable Name** and **Variable Type** properties are only available for array type list boxes.

The “Data Source” theme has different properties depending on the type of list box.

Data Source Theme (Selection Type List Box)

- **Expression**: The **Expression** lets you specify the 4D expression to be associated with the column:



You can enter:

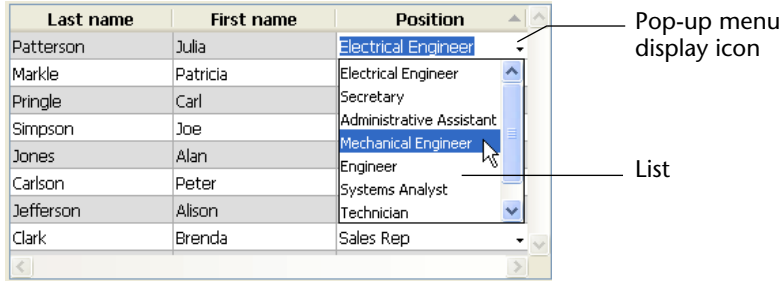
- A *4D expression* (simple expression, a formula or a 4D method). The result of the expression will be automatically displayed when you switch to Application mode. The expression will be evaluated for each record of the selection (current or named) of the Master Table. If it is empty, the column will not display any results.
- A *simple variable* (in this case, it must be explicitly declared for compilation). You can use any type of variable except for BLOBs and arrays. The value of the variable will be generally calculated in the On Display Detail event.
- A *field* using the standard [Table]Field syntax (example: [Employees]LastName). The following types of fields can be used:
  Alpha
  Text
  Numeric
  Date
  Time
  Picture
  Boolean
  You can use fields from the Master table or from other tables.
  In any case, you can specify the expression using the 4D Formula editor by clicking on the [...] button in the Property List:

If an expression is used, the column is never enterable even if the **Enterable** option is checked.
If a field or a variable is used, the column can be enterable or not depending on the **Enterable** option (and the "Can't Modify" attribute in the Structure editor).
In Design mode, the data source type is displayed in the first row of the column. For example, *Field=[Table1]MyFld*.
If the specified expression is not correct, the column of the list box will display an error message in Application mode.

- **Data Type**: This menu is used to define the type of expression or variable associated with the column. It is used to indicate the display format to apply and lets you update the **Display Type** menu in the "Display" theme.
  If a field is entered in the Expression area, the **Data Type** property is not displayed; the display format corresponding to the field type is used.

Data Source Theme (Array Type List Box)

- **Choice List**: This property can be used to associate a **choice list** with a column of the list box. If you designate a list, the user can use his or her values (displayed using a pop-up menu) to modify the values of the column and its associated array:



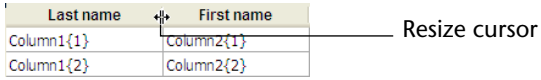
Note that keyboard entry is still possible. If you want for values to only be modified using the pop-up selection menu, choose a required list (see below).

Note If the specified list is hierarchical, only the items of the first level are taken into account.

Coordinates & Sizing Theme

This theme groups together the properties related to the list box column width.

- **Width**: Default column width (in pixels). This value is updated when you resize the column using the mouse in the Form editor.



If the **Resizable** property is checked, the user can also manually resize the column.

- **Minimum Width**: The minimum width of the column (in pixels). The width of the column cannot be reduced below this value when resizing the column or form.
- **Maximum Width**: The maximum width of the column (in pixels). The width of the column cannot be increased beyond this value when resizing the column or form.

Note When resizing the form, if the **Grow** horizontal sizing property was assigned to the list box, the right-most column will be increased beyond its maximum width if necessary.

Resizing Options Theme This theme only contains the **Resizable** option (checked by default). When this option is checked, the user can resize the column by moving the sides of the header area.

Entry Theme This theme groups together all the properties related to data entry in the list box column.

- **Enterable**: Authorizes column entry (checked by default). To change the value of a cell, the user must click twice on the value.
  When this property is disabled, any pop-up menus associated with the column via a list ("Data Source" and "Range of Values" themes) are disabled.
- **Entry Filter**: Associates an entry filter with column cells. This property is not accessible if the **Enterable** property is deselected.

Range of Values Theme This theme allows setting lists used to manage list box column entry.

- **Required List**: Allows setting a list where only these values can be inserted into the column. The list values are accessible using a pop-up menu associated with each cell (please refer to the "Choice list" property in the **Data Source** theme).
  Unlike the **Choice List** property, when a required list is defined, keyboard entry is no longer possible, only the selection of a list value using the pop-up menu is allowed. If different lists are defined using the **Choice List** and **Required List** properties, the **Required List** property has priority.

- **Excluded List**: Allows setting a list whose values cannot be entered in the column. If an excluded value is entered, it is not accepted and an error message is displayed.

Note If the specified list is hierarchical, only the items of the first level are taken into account.

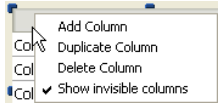
Display Theme

This theme is used to specify the display format for column values and the state of the **Invisible** property. The contents of this theme vary according to the type of variable set in the **Objects** theme.

- ***Type* Format**: Used to associate a display format with the column data. The formats provided will depend on the variable type (array type list box) or the data/field type (selection type list box). The standard 4D formats that can be used are: Alpha, Numeric, Date, Time, Picture and Boolean. The Text type does not have specific display formats. Any existing custom formats are also available.
- Boolean arrays can be displayed as check boxes or pop-up menus. If you choose the **Check Box** option, the **Title** property appears so that you can enter the title of the check box.
  If you choose the **Pop-up** option, the **Text when True** and **Text when False** properties appear, allowing you to set both titles for the pop-up menu.
- Columns with numeric values can be displayed as three-states check boxes. This option is found in the **Display Type** drop-down list. If you choose the **Three-states checkbox** type, the following values will be displayed:
  - an unchecked box represents the value 0,
  - a checked box represents the value 1,
  - a filled box (third state) represents any other value. For data entry, the value 2 is used for this state.

  In this case as well, the **Title** property appears so that the title of the check box can be entered.

- The **Invisible** property, when checked, allows hiding the column in the Application environment.
  In the Design environment, you can choose to display or hide the invisible columns using the list box contextual menu (click on a column or column header):



Background and Border Theme

This theme groups together the properties related to the custom background colors of column rows. By default, general list box background colors are used.

- **Background Color**: Allows setting a specific background color for the column.
- **Alternate Background Color**: Allows setting a specific alternate background color for odd-numbered rows of the column.

Text Theme

This theme groups together the specific properties related to text displayed in the column (style sheet, font, attributes, etc.). By default, the general text properties of the list box are used.

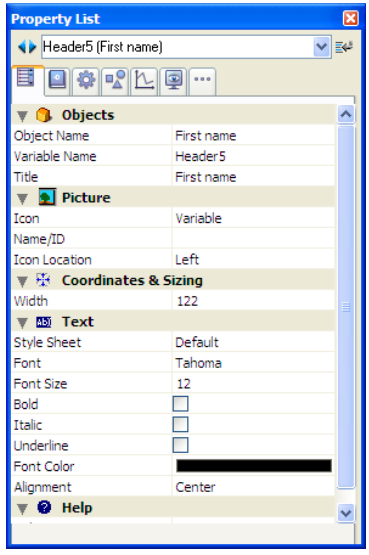
List Box Header Specific Properties

You can select a list box header in the Form editor by clicking it when the List Box object is selected:

Last name	First name	Position
Column1{1}	Column2{1}	Column3{1}
Column1{2}	Column2{2}	Column3{2}
Column1{3}	Column2{3}	Column3{3}
Column1{4}	Column2{4}	Column3{4}

Selected header

In this case, the Property List displays the following specific properties:



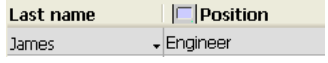
Objects Theme

This theme contains the properties used to define the header.

- **Object Name**: Name of the Header object.
- **Variable Name**: Name of the variable associated with the Header object. This variable (numeric) allows managing the current sort of the column and the display of the sort arrow programmatically (see the section “Management of List box objects” in the *4D Language Reference* manual).
- **Title**: Label appearing in the header.

Picture Theme

This theme contains properties that allow displaying a picture in the column header (optional). An icon can be displayed in the header next to or in place of the column title, especially when performing customized sorts.



- **Icon**: Defines the source of the picture to insert in the header. Like with 4D picture buttons, you can use a picture coming from a variable, picture library, a resource file or a file.

- **Name/ID**: Allows setting the picture to use in the previously defined source. The information to enter in this field depends on the selected source: name (if the picture is a variable), ID number (if the picture comes from the picture library or from a resources file) or pathname (if the picture comes from a picture file); in this latter case, the pathname must be relative to the Resources folder of the database (see the "Automatic Referencing of Picture Files" paragraph on page 456).
- **Icon Location**: Position of the icon in the header. You can place it to the **Left** or **Right** of the header.

Coordinates & Sizing Theme

This theme contains the **Width** property, whose value is the same as that set for the column.

Text Theme

This theme groups together the specific properties related to text displayed in the header (style sheet, font, attributes, alignment, etc.). By default, the general text properties of the list box are used.

Display of Fields in List Boxes

You can associate list box columns with fields from the master table and/or fields from different tables. For more information about the master table, please refer to the "List Box Specific Properties" paragraph on page 598.

However, in all cases, the contents of the list box will be based on the current selection (or a named selection) of the master table of the list box:

- If you only use fields from the master table, the contents of the list box rows will simply be modelled on those of the master table selection.
- If you use fields that do not belong to the master table, these "foreign" tables must be related to the master table by a Many-to-One relation, otherwise the "foreign" fields will be displayed empty. Automatic relations will be activated for each record of the master table selection and the list box will display the corresponding data in the related fields. If you use manual relations, you must program the activation of the relations in order to display the data in the list box.

  If an inconsistency in the definition of the list box causes columns to be displayed empty, an error message will appear in the Application mode in each incorrect column.

▼ We will use an example to explain the different cases.
Given a database with two tables: [Companies] and [Employees].

The current selection for the [Companies] table is the following:

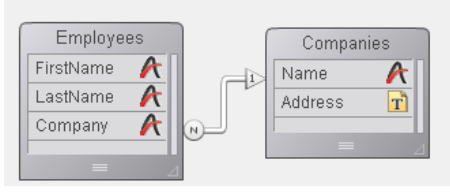
Company Name
Big Encyclopedias
Tiny Computers
Boring Travel Company

The current selection for the [Employees] table is the following:

First Name	Last Name	Company Name
Carla	Packard	Boring Travel Company
Andrew	Black	Tiny Computers
Vincent	Laughter	Boring Travel Company
Oliver	Dawson	Boring Travel Company
Sylvia	Fairview	Tiny Computers
Robert	Lanzel	Big Encyclopedias
Arnold	Schmitt	Boring Travel Company
Elizabeth	Jones	Big Encyclopedias
Yolanda	Court	Tiny Computers
Pascal	Pratt	Tiny Computers

The [Companies]Name field is associated with the first column of a list box. The [Employees]First Name and [Employees]Last Name fields are associated with the next two columns. The data source of the list box is the current selection.

■ **Case 1**: The two tables are related using an automatic relation.



1) The master table of the list box is [Employees]. The list box displays the current selection of the [Employees] table and activates the automatic relation to display the name of the company for each employee:

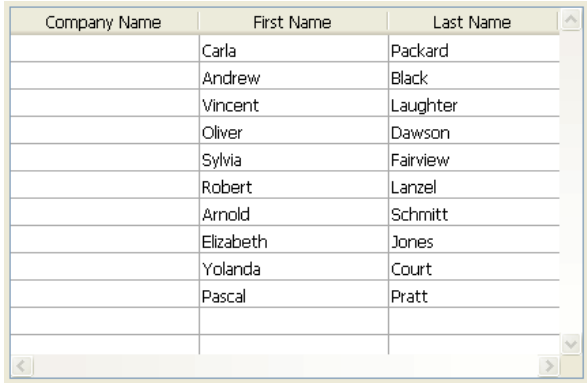
Company Name	First Name	Last Name
Boring Travel Company	Carla	Packard
Tiny Computers	Andrew	Black
Boring Travel Company	Vincent	Laughter
Boring Travel Company	Oliver	Dawson
Tiny Computers	Sylvia	Fairview
Big Encyclopedias	Robert	Lanzel
Boring Travel Company	Arnold	Schmitt
Big Encyclopedias	Elizabeth	Jones
Tiny Computers	Yolanda	Court
Tiny Computers	Pascal	Pratt

2) The master table chosen for the list box is [Companies]. The list box displays the current selection of the [Companies] table. Since there are only three records in this selection, only three rows are displayed in the list box. The columns for the [Employees]First Name and [Employees]Last Name fields are empty.

Company Name	First Name	Last Name
Big Encyclopedias		
Tiny Computers		
Boring Travel Company		

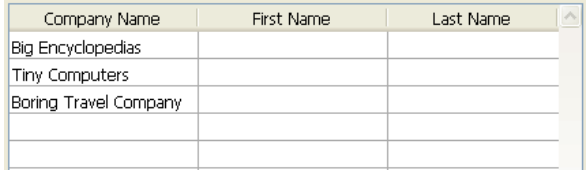
- **Case 2**: The two tables are not related (or they are related using a manual relation).

1) The master table of the list box is [Employees]. The list box displays the current selection of the [Employees] table. The column of the [Companies]Name field is empty.



Company Name	First Name	Last Name
	Carla	Packard
	Andrew	Black
	Vincent	Laughter
	Oliver	Dawson
	Sylvia	Fairview
	Robert	Lanzel
	Arnold	Schmitt
	Elizabeth	Jones
	Yolanda	Court
	Pascal	Pratt

2) The master table chosen for the list box is [Companies]. The list box displays the current selection of the [Companies] table. Since there are only three records in this selection, only three rows are displayed in the list box.
The columns for the [Employees]First Name and [Employees]Last Name fields are empty.



Company Name	First Name	Last Name
Big Encyclopedias		
Tiny Computers		
Boring Travel Company		

Naturally, you can manage the selections of different tables by programming and in this way display columns associated with fields that do not belong to the master table.

Displaying the Result of an SQL Query in a List Box

4D includes a powerful SQL engine that can be used to carry out queries on the data (see the 4D *SQL Reference* manual). It is possible to place the results of an SQL query directly in a list box. Only queries of the SELECT type can be used.

This works according to the following principles:

- Create the list box which will receive the query results. It is recommended to give the list box the same number of columns as there will be in the SQL query result (see below).
- The data source of the list box must be set to **Arrays**.

- Execute an SQL query of the SELECT type and assign the result to the variable associated with the list box. You can use the Begin SQL/End SQL tags (see the *4D Language Reference* manual).
- When the contents of a list box come from an SQL query, the columns cannot be sorted or modified by the user.
- Each new execution of a SELECT request with the list box leads to the resetting of the columns (it is not possible to fill the same list box progressively using several SELECT requests).

Note This mechanism cannot be used with an external SQL database.

▼ We want to retrieve all the fields of the PEOPLE table and put their contents into the list box having the variable name *vlistbox*. In the object method of a button (for example), simply write:

```
Begin SQL
   SELECT * FROM PEOPLE INTO <<vlistbox>>
End SQL
```

Using Object Methods with Fields and Objects

You can attach a method to any active object in a form. Methods that are attached to individual objects on a form are called *object methods*.

The following are some of the more common uses of object methods:

- Enforcing data entry constraints,
- Initializing and managing interface objects such as tab controls, pop-up menus, drop-down lists, list boxes, combo boxes, hierarchical lists, and pop-up menus.
- Specifying the action that takes place when an object is clicked or double-clicked,
- Managing drag-and-drop operations.

The previous section on types of active objects gives several simple examples of how methods are used to manage various kinds of objects. Following are some simple examples that perform operations on data.

The following method calculates a total based on data in two other fields:

```
Line Total:= [Products]Price * [Orders]Quantity
```

Here is a method to make all characters in a Name field uppercase:

[Customers]Name:= **Uppercase** ([Customers]Name)

Note By default, built-in functions are displayed in boldface and user-written functions are displayed in italics. For a list of built-in functions, see the *4th Dimension Language Reference* manual.

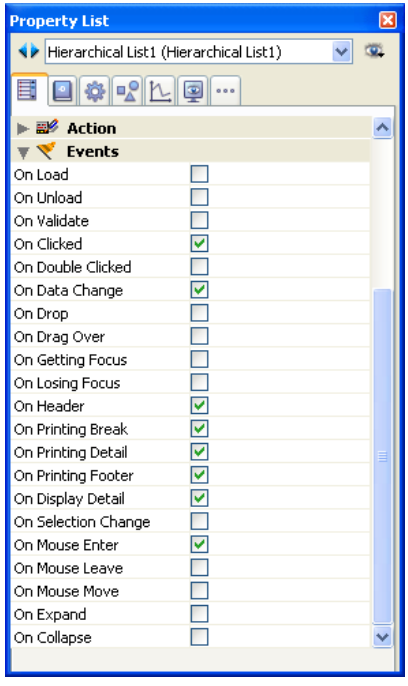
The following method concatenates values from a First Name field and a Last Name field and assigns the results to a variable named vName:

vName:= [Employees]First Name + " " + [Employees]Last Name

Because each object method is attached to its object, you create object methods from within the Form editor. For information on how to use methods, see chapter 11 “Methods” on page 757.

Object Events

Object methods run when certain events occur. For example, the action associated with a tab control makes sense only when a user clicks a tab. In a scrollable area, you may want the method to execute only when the user double-clicks an item. You can specify which events will execute the object method for a particular object in the “Events” section of the Property List



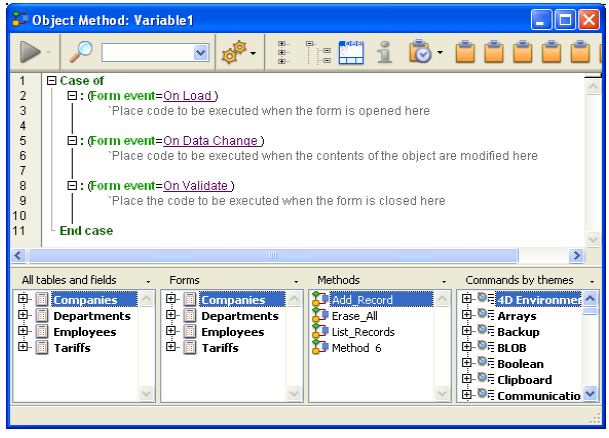
The list contains all the form events that are pertinent for the selected object. Some events are only available for specific types of objects. There are also additional events that are only generated at the form level.
For a detailed description of all form events, refer to the Form event command in the *4D Language Reference* manual.

You select an event by clicking on the associated check box. To select or deselect all the events at once, press the **Ctrl** key (Windows) or the **Command** key (Mac OS) while clicking any event.

If you need to execute different code segments for several different events, use a Case statement in your method and test for each event you checked in the Events section. To test for an event, you use the Form event function and the Form Event constants on the Constants page of the Explorer.

You can add a constant to your code by opening the Explorer to the Constants page and dragging the desired constant to the desired point in the code. By default, all constants are underlined in the Method editor.

An example shell for an object method might look like this:



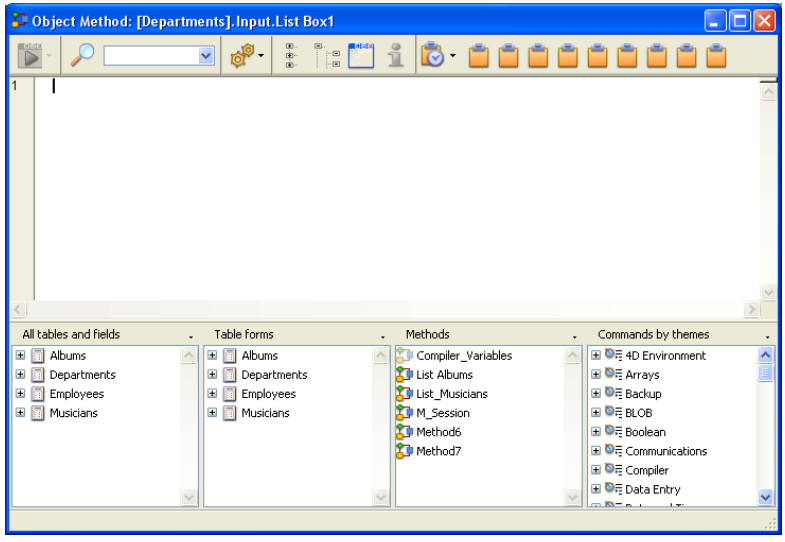
For more information on object methods, see the "Where to Put an Object Method" paragraph on page 765 and the "Form Events" section in the *4D Language Reference* manual.

► To add an object method:

1 **In the Form editor, select the object to which you want to assign a method.**

2 Click the Edit button located next to the Object Method line ("Action" theme) in the Property List.
OR
Choose Object Method from the Object menu.
OR
Click the object using the right mouse button and choose Object Method from the contextual menu.
OR
Hold down the Alt key (Windows) or Option key (Mac OS) and click the field or object.

4D displays a new Method editor window, blank by default. The method name is "Object Method:" followed by the form name then the object or field name.



3 Write the method.

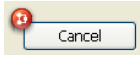
You can use several techniques to enter text into the Method editor. You can:

- Type text into the editor or type the first few letters and then hit the **Tab** key in order to use the type-ahead function,
- Double-click on field or table names, methods, commands or macros in the scrollable areas below the text area,
- Drag table names, field names, form names, constants, built-in commands, plug-in commands, or project methods into the editor from the Explorer.

For more information on how to use the Method editor, see the "Creating or Opening Methods" paragraph on page 765.

4 **Close the Method window (optional).**

The method is now associated with the field or active object. You can display the objects with an associated method by displaying the "Object Method" shields:



You can view or modify a method at any time.

► To open a method for viewing or modification:

1 **In the Form editor, select the object whose method you want to open.**

2 **Click the Edit button located next to the Object Method line ("Action" theme) in the Property List.**
OR
Choose Object Method from the Object menu.
OR
Click the object using the right mouse button, then choose Object Method from the contextual menu.
OR
Hold down the Alt key (Windows) or the Option key (Mac OS) and click the object to which the method is attached.

The Method editor appears with your method, ready for you to make any changes.

Deleting an Object Method

You can delete an object method at any time. To do so, select the object to which the method is attached and then choose **Clear Object Method** from the **Object** menu.

Adding a Subform to the Form

A subform is a form, usually of the List type, from another table or subtable in the Master table that is displayed in a Detail form. A subform lets you enter, view, and modify data in other tables. You usually use subforms in databases in which you have established One to Many relations. A subform on a form in a related One table lets you view, enter, and modify data in a related Many table. You can have several subforms coming from different tables in the same form. However, it is not possible to place two subforms that belong to the same table on the same page of a form.

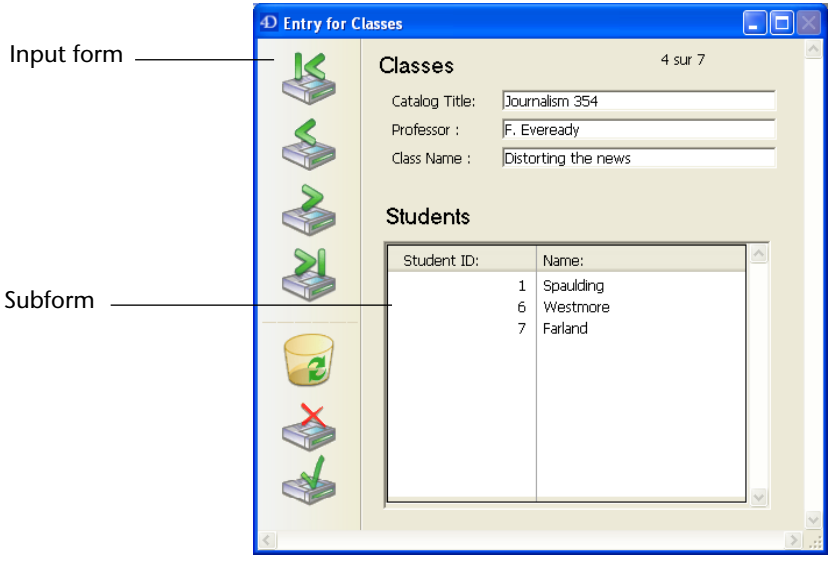
For example, a Contacts manager database might use a subform to display all the telephone numbers for a particular contact. Although the telephone numbers appear on the Contacts screen, the information is actually stored in a related table.

Using a One to Many relation, this database design makes it easy to store an unlimited number of telephone numbers per contact. With automatic relations, you can support data entry directly into the related Many table without programming. For more information about relations, refer to the "Relating Tables" paragraph on page 295.

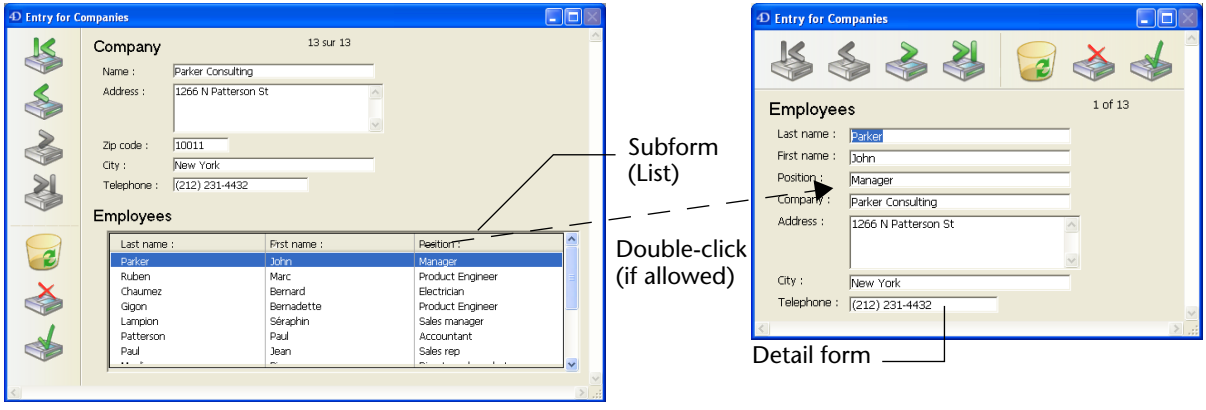
Although subforms are generally associated with Many tables, a subform can display the records of any other database table.

Similarly, subforms generally display data as a list, but it is also possible to display data as a page. This point is described in the "Subforms in Page Mode" paragraph on page 628.

You can create a subform using the Form Wizard when creating a new form, or add one to an existing form using the Form editor. You must create the List form that you want to use as the subform beforehand.

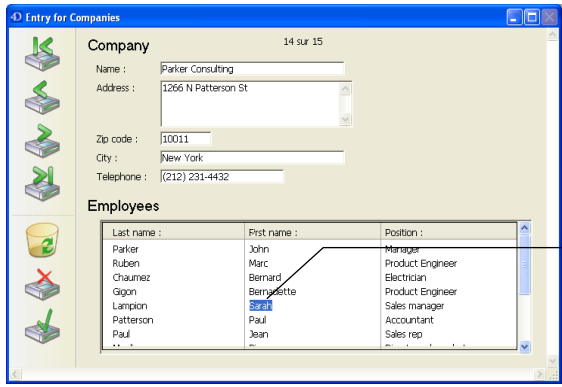


Subforms can be used for data entry in two ways: the user can enter data directly in the subform, or enter it in an input form. The figure below shows both a subform and the input form associated with it.



In this configuration, the form used as the subform is referred to as the *List form*. The input form is referred to as the *Detail form*.

You can allow the user to enter data in the Detail form and/or in the List form:



Creating a Subform

You can add a subform to a form in three ways:

- In the Form Wizard, using the Subform page in the Advanced options,
- In the Form editor, using the Subform tool of the object bar,
- In the Form editor, using the Forms page of the Explorer.

Of course, you can mix these different techniques according to your needs.

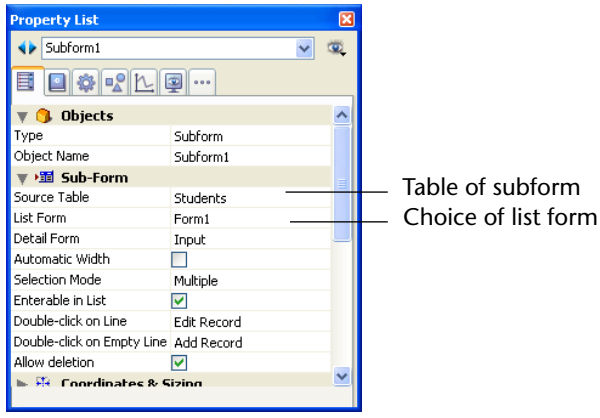
This section describes adding and configuring a subform from the Form editor. For more information about creating subforms using the Form Wizard, refer to the "Subform Page" paragraph on page 363.

▶ To create a subform using the Subform tool:

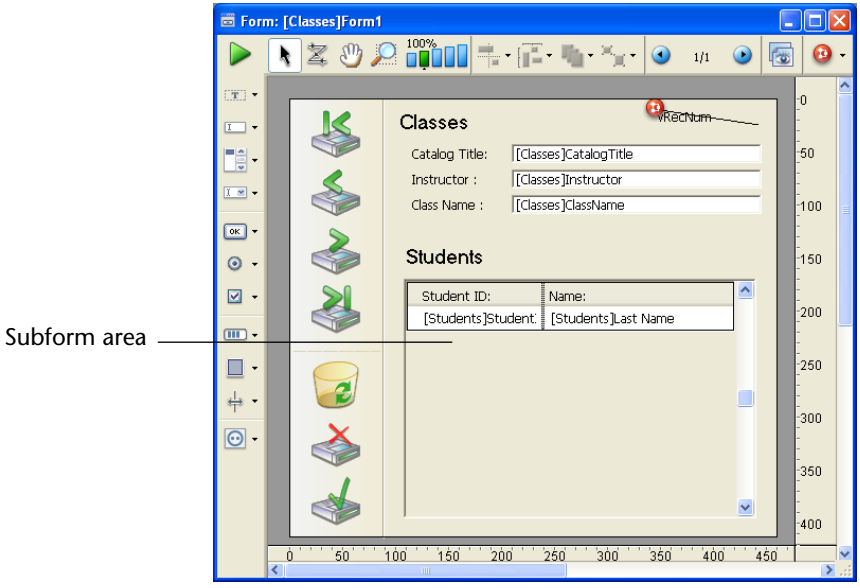
1 **Create and configure the form that you want to use as the subform.**
For more information about list forms, refer to chapter 8 "Output Displays and Reports" on page 635.

2 **In the input form, create a subform object using the Subform tool of the object bar (variation of the last button).**

3 **In the Property List ("Sub-Form" theme), choose the Source Table and the List Form that you want to use.**



The subform then appears in the form. You can resize and reposition it as desired.



4 **(Optional) Specify the detail form that you want to associate with the subform.**

This is described in the "Associating a Detail Form with a Subform" paragraph on page 627.

5 **If necessary, add buttons for managing subrecords and set the options concerning subform operation.**

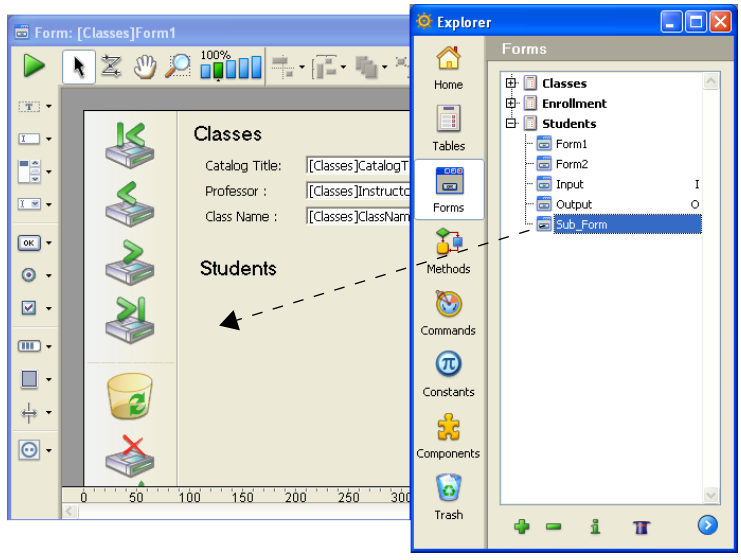
To do this, refer to the "Subforms Options" paragraph on page 629.

► To add a subform via the Explorer:

1 **Create and configure the form that you want to use as the subform.**
For more information about list forms, refer to chapter 8 “Output Displays and Reports” on page 635.

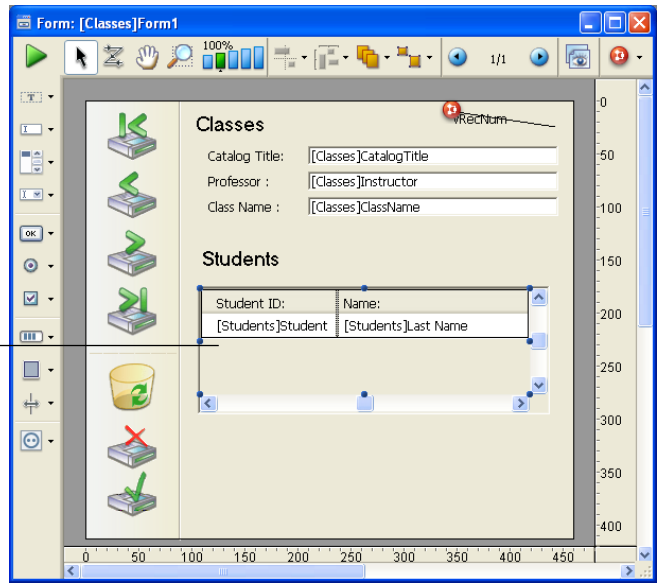
2 **Open the Forms page of the Explorer and expand the table that contains the subform you want to insert.**

3 **Drag the List form from the Explorer to the main form.**



Note You can create a subform object in the input form beforehand and drop the form on it. This lets you predefine the size of the subform.

The subform then appears in the form. You can resize and reposition it as desired.



Note You can open the source form directly in a new window of the Form editor by selecting the **Edit Form...** command in the contextual menu (**right-click** on the subform).

4 **(Optional) Specify the detail form associated with the subform.**

This is described in the "Associating a Detail Form with a Subform" paragraph on page 627.

5 **If necessary, add buttons for managing subrecords and set the options concerning subform operation.**

To do this, refer to the "Subforms Options" paragraph on page 629.

By default, 4D places scroll bars so that the user can scroll through the data of the subform. These scroll bars can be removed (see the "Scroll Bars" paragraph on page 633).

Associating a Detail Form with a Subform

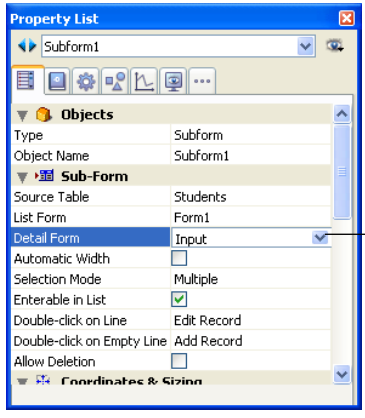
You can associate a detail form with a list subform. The detail form can be used to enter or view subrecords. It generally contains more information than the subform. Naturally, the detail form must belong to the same table as the subform.

You normally use an Output form as the list form and an Input form as the detail form. If you do not specify the form to use for full page entry, 4D automatically uses the default Input format of the table.

Depending on the configuration of the subform, the user may display the detail form by double-clicking on a subrecord or by using the commands for adding and editing subrecords.

► To associate a detail form with a subform:

1 Specify the detail form in the Property List:



OR

Hold down Shift and drag the detail form from the Explorer onto the subform.

You can change the list or detail forms of the subform at any time by repeating the steps described above.

Adding Buttons for Managing Subrecords

You can add custom buttons for handling data entry in a subform. Any kind of button — standard, highlight or invisible — can be used. For example, the Form Wizard adds the following buttons if the option is checked:

Buttons added by Form Wizard

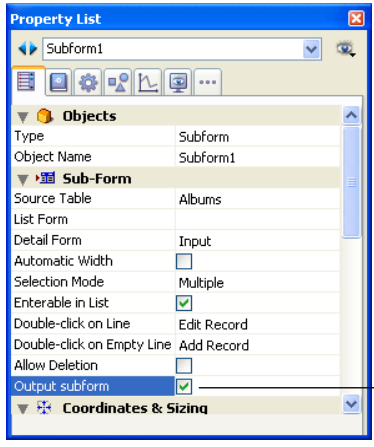
You can set the action of these buttons using programming (see the "Subrecords" section of the *4D Language Reference* manual) or via standard actions.
4D offers three standard actions to meet the basic needs for managing subrecords: **Edit Subrecord**, **Delete Subrecord**, and **Add Subrecord**. When the form includes several subforms, the action will apply to the subform that has the focus.

For more information about adding these buttons and their associated standard actions, refer to the "Buttons" paragraph on page 551.

Subforms in Page Mode

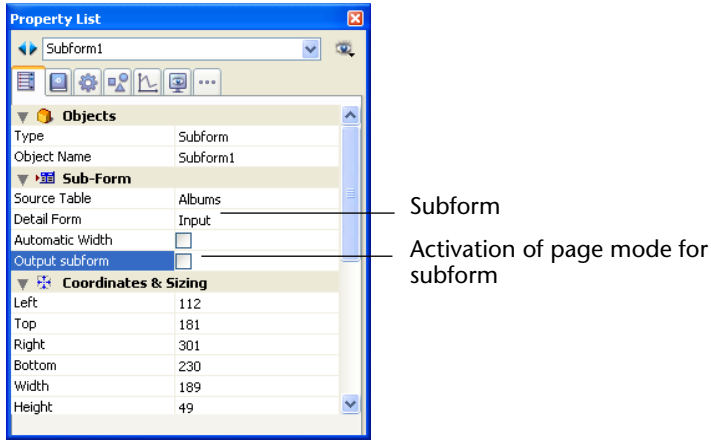
It is possible to use subforms in the form of pages, as with the DIALOG command. In this case, the subform can display the data of the current subrecord or any type of pertinent value depending on the context (variables, pictures, and so on). It is also possible to use this subform for data entry.

The **Output subform** subform property can be used to configure this functioning:



When this option is checked, the subform is used in list mode (standard mode). By default, this option is checked.

To activate the page mode, simply uncheck this option. In this case, the properties related to the configuration of subforms as lists (Selection Mode, Double-click on Line, and so on) are no longer displayed:



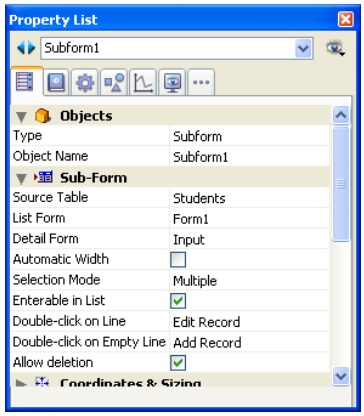
The subform in page mode uses the input form indicated by the “Detail Form” property. Unlike a subform in list mode, the form used can come from the same table as the parent form. It is also possible to use a project form.

During execution, a subform in page mode has the same standard display characteristics as an input form. The output form mechanisms (related more particularly to the management of markers) are not activated.

Subforms Options

You can set several specific properties related to subforms. These properties concern the width of the subform as well as the user actions allowed (selection and entry modes).

Most of the specific subform properties are placed under the "Sub-Form" theme of the Property List:



Automatic Width

You can resize the subform area in the form as desired, just like any other form object.
You can also let 4D automatically set the width of the subform so that all the fields it contains are displayed. To do this, check the **Automatic Width** option in the object properties.

Note When you create a subform area by dragging a form directly from the Explorer, 4D automatically sets the width of the subform.

Selection Mode

List subforms can have three user selection modes: **None**, **Single** and **Multiple** (these options are not available for subforms in page mode).

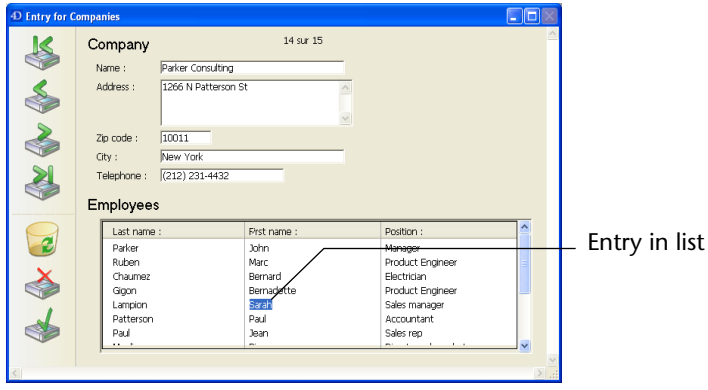
- **None**
  Records cannot be selected if this mode is chosen. Clicking on the list will have no effect unless the **Enterable in List** option is checked.
  The navigation keys only cause the list to scroll; the On Selection Change form event is not generated.

- **Single**
  One record at a time can be selected in this mode. Clicking on a record will select it and it will become the current record. A **Ctrl+click** (Windows) or **Command+click** (Mac OS) on a record toggles its state (between selected or not).
  The **Up** and **Down** arrow keys select the previous/next record in the list. The other navigation keys scroll the list. The On Selection Change form event is generated every time the current record is changed.

- **Multiple**
  Several records can be selected simultaneously in this mode. The selected subrecords are returned by the GET HIGHLIGHTED RECORDS command. Clicking on the record will select it, but it does not modify the current record. A **Ctrl+click** (Windows) or **Command+click** (Mac OS) on a record toggles its state (between selected or not). The **Ctrl+click** (Windows) or **Command+click** (Mac OS) and **Shift+click** key combinations allow you to make multiple selections.
  The **Up** and **Down** arrow keys select the previous/next record in the list. The other navigation keys scroll the list. The On Selection Change form event is generated every time the selected record is changed.

Enterable in List

When a list subform is **Enterable in List**, the user can modify record data directly in the list, without having to use the associated detail form. To do this, simply click twice on the field to be modified in order to switch it to editing mode (make sure to leave enough time between the two clicks so as not to generate a double-click). By default, this mode is activated for all list subforms.



When this option is not checked, entry must be carried out via the associated detail form.

Action in Event of Double-click

You can set parameters for how a list subform should behave in response to a user double-click.
In databases created with a previous version of 4D, you can also set the response to a double-click on an empty line.

- **Double-click on Line**: Action to perform in case of a double-click on a subform record. The following options are available:
  - **Do nothing**: Ignores double-click.

- **Edit Record**: Changes the subform record to editing mode. Modification will be carried out directly in the list if the "Enterable in List" option is checked. Otherwise, it will be carried out in page mode, in the detail form associated with the subform.
- **Display Record**: Displays the data of the record in page mode in the detail form associated with the subform (read only).

- **Double-click on Empty Line (converted forms only)**: In previous versions of 4D, adding records to a subform could be carried out by double-clicking on an empty line of the subform. For compatibility reasons, this mechanism is kept for forms created with previous versions of 4D — however, it is now recommended to use standard actions to add records. You can set the action to perform in case of a double-click on an empty line of the subform. The following options are available:
  - **Do nothing**: Ignores double-click.
  - **Add Record**: Creates a new record in the subform and changes to editing mode. The record will be created directly in the list if the "Enterable in List" option is checked. Otherwise, it will be created in page mode, in the detail form associated with the subform.

Allow Deletion

By default, the user can delete subrecords in a list subform using the **Del** or **Backspace** keys.
Since this could disturb the standard operation of the interface for certain applications (based, for instance, on buttons), you can prevent it using the **Allow deletion** option.
When this option is not checked, the user can no longer delete subrecords using the deletion keys of the keyboard.

Focusable

The subform object can have the **Focusable** property ("Entry" theme). When a subform has the focus, the user can control it using navigation keys, using the **Select all** menu (if the selection has multiple lines), etc.

When a subform receives or loses the focus, the form method of the host form is called using the On getting focus or On losing focus events. In this case, the Focus object command returns a pointer to the table or subtable of the subform (for more information about this command, refer to the *4D Language Reference* manual).

As with all focusable objects, the **Hide focus rectangle** property can be used to not show the focus graphically.

Scroll Bars

By default, 4D adds vertical and horizontal scroll bars to subforms in order to facilitate navigation and display of information. These bars are added to the subform, outside of the original subform area. You can remove them by deselecting the **Horizontal Scroll Bar** and/or **Vertical Scroll Bar** options in the "Appearance" theme.

Printing

Since there may be several records that the subform area cannot hold, 4D offer three options ("Print" theme) to handle the printing of subform records:

- **Variable** (default option)
  If you check this option, 4D enlarges or reduces the subform area in order to print all the subrecords.
- **Fixed (Truncation)**
  If you check this option, 4D only prints the subrecords that appear in the subform area. The form is only printed once and the records not printed are ignored.
- **Fixed (Multiple Records)**
  If you check this option, the initial size of the subform area is kept but 4D prints the form several times in order to print all the records.

Notes

• You cannot place any objects on either side of a variable frame subform. Objects placed on either side of the subform will be repeated for every line of the subform.
• You cannot print more than one subform coming from the same table on the same form page.
• For more information about printing objects of variable size, refer to the "Printing Subforms, Pictures, and Text Fields" paragraph on page 677.

8 Output Displays and Reports

Output forms are used for two purposes: listing records on screen and printing reports. This chapter explains the following operations:

- Using the Form Wizard to create output forms for listing records on-screen,
- Using the Form Wizard to create forms for output forms and printed reports,
- Using the Form editor to customize output forms and printed reports.

In numerous cases, you can create a report more quickly using the Quick Report editor. However, the Form editor gives you more customized control over the final appearance of your report. For more information about using the Quick Report editor, refer to the chapter "Quick Reports" on page 699.

Note Only table forms can be used as output forms. Project forms are intended to be used as detail forms. For more information about these types of forms, please refer to the "Table Forms and Project Forms" paragraph on page 339.

Output Forms for Listing Records

A form that displays records as a columnar list contains separate areas:

- **Header area**: Contains the report title, column headers, and form management buttons or objects,
- **Detail area**: Contains the body of the report,

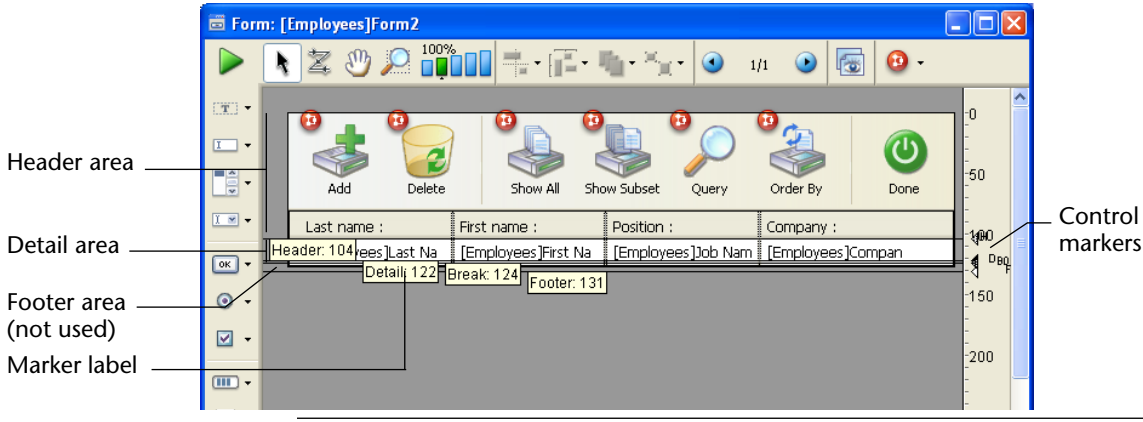
- **Footer area**: Contains buttons or summary calculations based on all the records in the report,
- **Break area**: Contains text or graphics that appear after the list of records and summary calculations based on all the records or sub-groups of records.

When you create a List form using the Form Wizard, it automatically creates these areas for you. It places the form title (the table name) and the field names in the Header area as well as the control buttons[1]. The fields you select are placed in a row in the Detail area. A small Break area is created but the Form Wizard puts nothing in it.

When you open the form in the Form editor, you can modify the size of each of these areas, modify the contents of any area, add objects to the Break area, and create additional Break areas for summary calculations.

The areas of the form that function as the Header, Detail, Break, and Footer areas are controlled by *output control lines*. By dragging the output control lines vertically, you can change the size of each area.

The following illustration shows an output form that was created using the Basic screen of the Form Wizard (XP template).

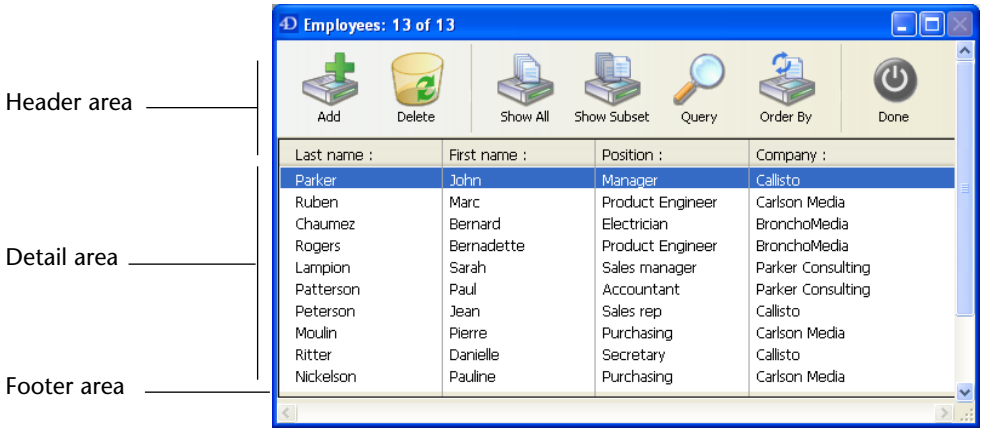


Note You can choose to hide/display markers and their labels. For more information, refer to "Showing/Hiding Elements in the Form Editor" on page 374.

1. For more information about using break areas, please refer to the "Reports with Breaks" paragraph on page 666.

The horizontal lines divide the report into Header, Detail, Break, and Footer areas. The area from the top of the form to the Header line is the Header area. Similarly, the area between the Header and Detail lines is the Detail area, and the Footer area extends from the top of the Break line[1] (labelled B0) to the Footer line.
You adjust the sizes of each area by dragging the Header, Detail, Break, or Footer markers, or their labels, vertically.

When this form is used, it looks like this:



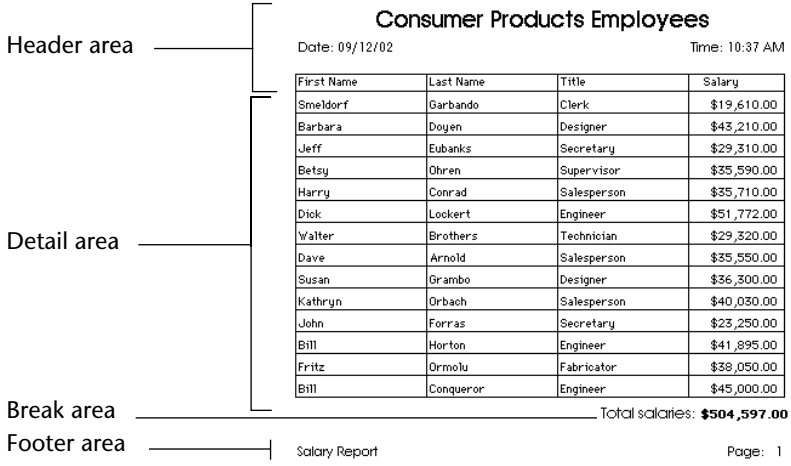
The Detail area expands dynamically as the window is resized, while the Header and Footer areas remain a fixed size.

In a form that lists records on screen, the Header and Footer areas can include clickable and non-enterable objects such as buttons, radio buttons, hierarchical lists, and so on. For more information, refer to the following paragraph concerning output control lines.

In a printed report, a Header area often contains the date, the time, and a running title as well as column labels. Records appear in the Detail area. A calculated total may appear in the Break area. The Footer area contains the page number.

1. For more information on Break areas refer to “Reports with Breaks” on page 666.

The following illustration identifies the different areas as they appear in a printed report.



A report may have additional Break areas for subtotals and other calculations. A report may also have additional Header areas that appear within the body of the report. The additional Header areas are used to identify subgroups. For an example of a report with several Header and Break areas, see the section "Creating Additional Control Lines" on page 669.

Using Output Forms

An output form lists records. Although any form can be used as an output form, most output forms have these features:

- Each row is a record.
- Each column is a field or a variable.
- Each column is labeled at the top of the window. The columns can be resized using the splitters between each title area.
- The header and/or footer of the form may contain buttons, pop-up menus, etc.

The following is a typical output form.

There are five basic ways to work with output forms:

- Scrolling to locate records
- Resizing columns
- Highlighting to select records
- Double-clicking to view a record
- Entering and modifying data in fields.

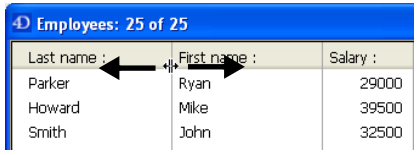
Scrolling

An output form has standard horizontal and vertical scroll bars. You can scroll records by clicking either of the scroll arrows, clicking the scroll bar, or dragging the scroll box. You can also use the **PgUp** and **PgDn** keys (to scroll through the list of records one screen "page" at a time) or the **Start** and **End** keys (to go directly to the start or end of the list).

If the output form has more fields than can be displayed in the window, you can scroll horizontally using the scroll bar at the bottom of the window.

Resizing Columns

You can resize the output form columns (except when this possibility has been removed in the form editor). To do so, place the mouse cursor over the title area of the columns; the cursor changes as follows ⇹ in order to indicate that the column can be resized. You can then click and move the edge of the column in order to reduce or enlarge it:



Highlighting

You highlight (select) a record by clicking it. You can highlight a single record, two or more adjacent records, or two or more non-adjacent records. After highlighting records, you can perform operations such as displaying them as a subset or deleting them.

► To highlight a record:

■ **Click a record displayed in the output form or press the ↑ or ↓ keys.**

The record you clicked is highlighted. If you use the arrow keys, the first or last record of the list is highlighted.

► To highlight several adjacent records:

1 **Click the first record you want to highlight.**

2 **Hold down the Shift key and click the last record you want to highlight (or press the ↑ or ↓ keys).**

All records between the first and last ones you clicked are highlighted.

► To highlight several non-adjacent records:

1 **Click the first record you want to highlight.**

2 **Hold down the Ctrl key (Windows) or Command key (Macintosh) and click another record.**

3 **Continue clicking records while holding down the Ctrl key (Windows) or Command key (Macintosh).**

Each record you click is highlighted.

► To highlight the entire current selection of records in the output form:

■ **Choose Select All from the Edit menu or use the standard Ctrl+A (Windows) or Command+A (Mac OS) shortcut.**

All records displayed in the output form are highlighted. To highlight all the records in the table, choose **Show All** from the **Records** menu (or the corresponding icon in the toolbar) before choosing **Select All** from the **Edit** menu.

Double-clicking

To view a record in the input form:

■ **Double-click the record in the output form.**

When you double-click a record, the input form replaces the output form in the frontmost window. The selected record is displayed in the input form, ready for modification.

Adding and Modifying Records

You can add and modify records directly in an output form. The output form is especially useful for modifying a few adjacent records because several records are displayed on the screen simultaneously.

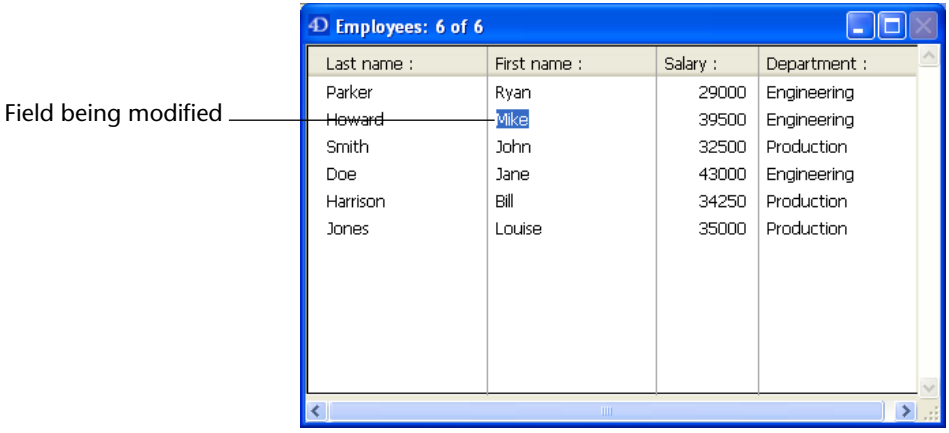
You can only enter or modify fields in the current table. You cannot enter or modify data in variables, fields from other tables, or subforms.

In the output form, you can:

■ Select and edit fields by clicking on them (you have to click twice if the record is not already selected),

■ Move from field to field and record to record using the **Tab**, **Enter** or **Return** keys,

- Add a new record, using the **Records** > **New Record in List** menu command:



Hitting the **Tab**, **Enter** or **Return** keys, or clicking on another field saves your changes.

Output Control Lines

You control the Header, Detail, Break, and Footer areas with the output control lines in the Form editor. You move the control lines vertically to allow more or less space for each area. Any object that you place in these areas is displayed or printed at the appropriate location.

The following explains how these areas work when the form is displayed or printed in Application mode or via the integrated functions of the Design mode:

- **Header area**: The Header area is displayed at the top of each screen and is printed at the top of each page of a report. The Header area is defined as the area above the Header control line (H). You make the Header area smaller or larger by dragging the Header control marker vertically. You can use the Header area for column names, for instructions, additional information, or even a graphic such as a company logo or a decorative pattern.

  You can also place and use active objects in the Header area of output forms displayed as subforms, in the records display window or using the DISPLAY SELECTION and MODIFY SELECTION commands.

All active objects can be inserted. This includes:

- Buttons, 3D buttons, highlight buttons, picture buttons,
- Combo boxes, pop-up menus/drop-down lists, picture pop-up menus and hierarchical pop-up menus,
- Scrollable areas, hierarchical lists, list boxes,
- Radio buttons, 3D radio buttons, picture radio buttons,
- Check boxes, 3D check boxes,
- Thermometers, rulers, dials.

Standard actions such as **Add Subrecord**, **Cancel** (lists displayed using DISPLAY SELECTION and MODIFY SELECTION) or **Automatic splitters** can be assigned to the inserted buttons. The following events apply to the active objects you insert in the header area: On Load, On Clicked, On Header, On Printing Footer, On Double Clicked, On Drop, On Drag Over, On Unload. Keep in mind that the form method is called with the On Header event after calling the object methods of the area. For more information on the MODIFY SELECTION and DISPLAY SELECTION commands, refer to the *4D Language Reference* manual.

- **Detail area**: The Detail area is displayed on the screen and printed once for each record in a report. The Detail area is defined as the area between the Header control line and the Detail control line (D). You make this area smaller or larger by dragging the Detail control marker vertically. Whatever you place in the Detail area is displayed or printed once for each record. Most often you place fields or variables in the Detail area so that the information in each record is displayed or printed, but you can place other elements in the Detail area as well.
- **Break areas**: Break areas are displayed once at the end of the list of records and are printed once after the records have been printed in a report. In the report above, the Break area is defined as the area between the Detail control line and the Break control line (labeled B0). There can be other Break areas in your report. You make Break areas smaller or larger by dragging the Break control marker vertically. You can use a Break area to display information that is not part of the records (instructions, current date, current time, etc.), or to display a line or other graphic element that concludes the screen display. In a printed report, you can use a Break area for calculating and printing subtotals and other summary calculations.

- **Footer area**: The Footer area is displayed on screen under the list of records. It is always printed at the bottom of every page of a report. The Footer area is defined as the area between the Break control line (B0) and the Footer control line (F). You make this area smaller or larger by dragging the Footer control marker vertically.
  You can use the Footer area to print graphics, page numbers, the current date, or any text you want at the bottom of each page of a report. For output forms designed for use on screen, the Footer area typically contains buttons that give the user options such as doing a search or sort, printing records, or putting away the current report. All active objects are accepted.

  Whenever any form is used for output, either for screen display or printing, the output control lines take effect and the areas display or print at designated locations. The output control lines also take effect when a form is used as the List form in a subform area.

  The output control lines have no effect when a form is used for input.

  Methods that are associated with objects in these areas are executed when the areas are printed or displayed as long as the appropriate events have been activated. For example, a object method placed in the Header area is executed when the On Header event takes place.

  You can create additional control lines to set additional Break areas and Header areas for a report. These additional areas allow you to print subtotals and other calculations in a report and to display other information effectively. Additional control lines are discussed in the section "Reports with Breaks" on page 666.

  Working with the output control lines and the areas they define are described in detail in subsequent sections of this chapter.

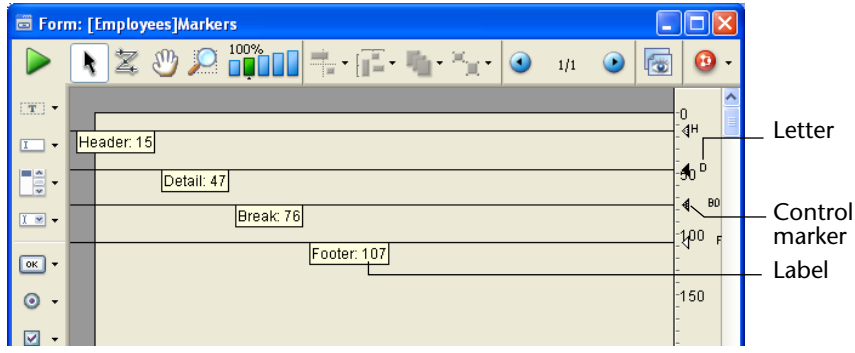
Moving Output Control Lines

You adjust the size of the Header, Detail, Break, and Footer areas by moving the output control markers.

Output control lines are displayed as lines across the form. Each control line has an identifying marker and label that is displayed in the ruler. The control marker is the triangle in the ruler and the label is the letter or letters next to the marker. You can move a control line by dragging its marker or label. By default, the labels of the control lines are always displayed; however, you can hide them if desired (refer to the "Showing/Hiding Elements in the Form Editor" paragraph on page 374). When they are hidden, you can display the labels temporarily by clicking on the control markers.

Labels indicate the name and location of each marker with respect to the form origin. When you move a marker, the label indicates the new location of the marker in real time. Labels allow you to move control lines even when the rulers are not displayed.

The figure below identifies control markers and labels:



To move a control line, drag the control marker or the marker label vertically.

- Holding down the **Shift** key while dragging a control marker moves all control lines below that control marker. For example, to drag all control lines together, hold down **Shift** and drag the Header marker. To move all control lines except the Header control line, **Shift+drag** the Detail marker.
- If you want to move objects located below the control marker or enlarge objects above the control marker while you are moving it, select each object you want to modify before beginning to move the control marker.

The control lines cannot be dragged out of order. For example, if you attempt to drag a Footer control line higher than a Break control line, the drag operation automatically stops when the Footer marker reaches the Break marker.

You can place markers and control lines on top of one another. Placing one marker on top of another reduces its area to nothing, removing it from the report. For example, if you have nothing to print in a Break area, you can drag the Break marker on top of the Detail marker. Doing so prevents 4D from creating space for a Break area. The report can thus utilize all the space available on the page.

If you don't want to print any details, drag the Detail marker on top of the Header marker. If you don't need a Header, drag the Header marker to the very top of the form (at point 0).

Active objects (fields or variables) located in the details of forms must not overlap the header or footer area, otherwise they will not be displayed when the form is executed.

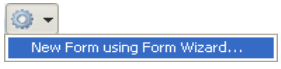
Creating Output Forms

You use the Form Wizard to create an output form for each table in your database. The principle is similar to that used for creating input forms. You simply select a different form type. If you use the Wizard's Advanced screen, a different set of options is available.

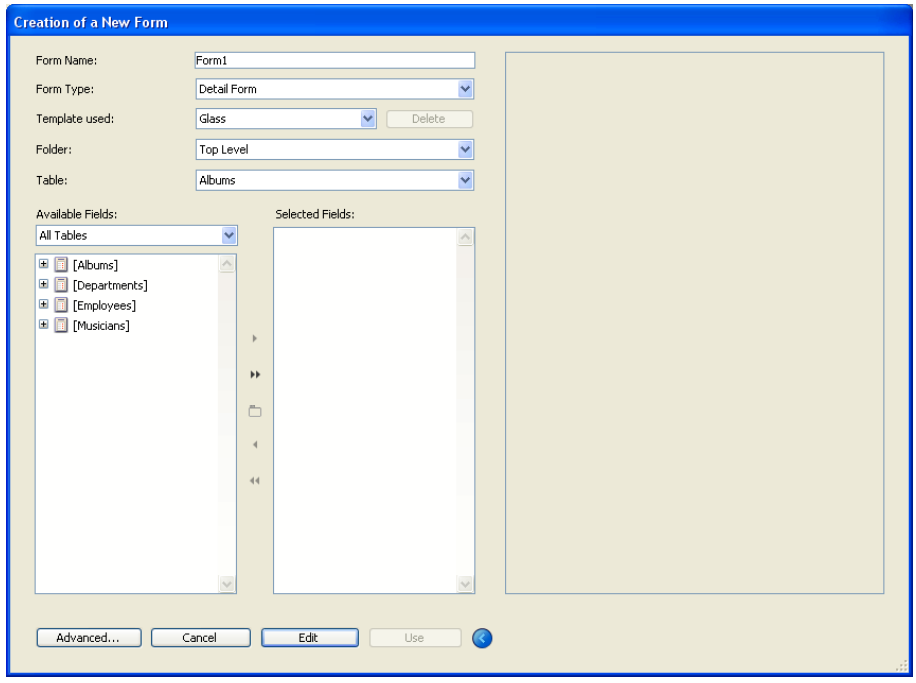
As with input forms, you can create an output form using either the Basic or Advanced screens of the Form Wizard.

► To create an output form:

1 **Choose New > Form... in the File menu or using the New button of the toolbar.**
OR
On the Forms page of the Explorer, select the name of the table in the Table Forms list for which you want to create a form then click on the options menu and choose New Form using Form Wizard.



The Basic screen of the Form Wizard appears.



If necessary, use the Table drop-down list to select the master table for the form.

The list of fields changes to reflect your selection.

2 **Name the form by filling in a name in the "Form Name" area.**

You can refer to the form by name using the language.

3 **Choose List form from the "Form Type" drop-down list.**

This selection instructs the Wizard to place the output control lines in the correct locations for lists, place the fields in a row in the Detail area, and add control buttons to the Header area.

Note This option is only available if you create a table form. Project forms cannot be used as list forms.

4 **Choose a template that is appropriate for output forms.**

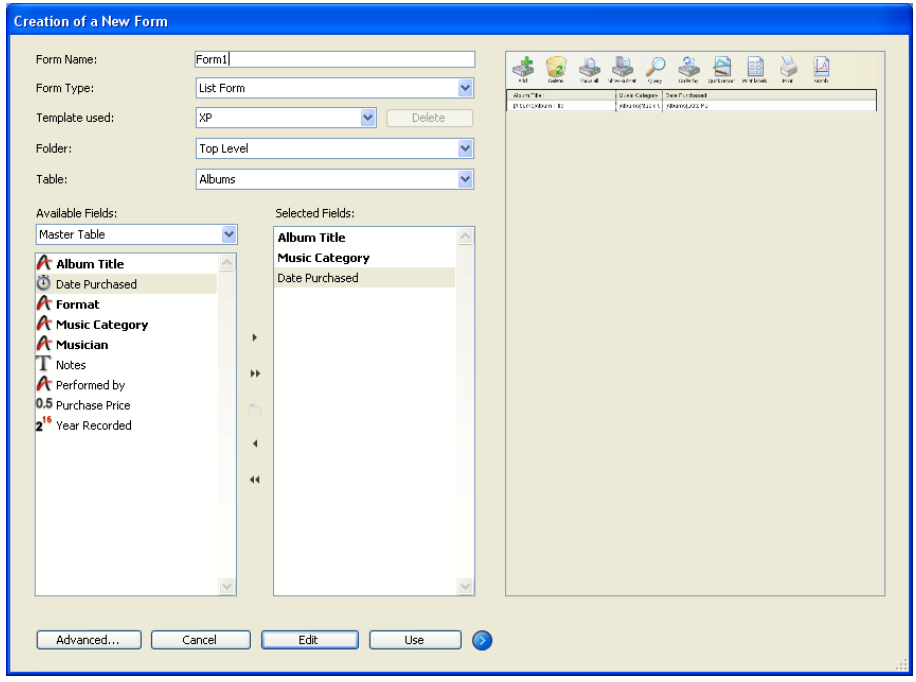
The template controls several aspects of the appearance of the form, including font attributes, field label placement, and the design of decorative rectangles surrounding the fields. 4D ships with several templates. You can also create custom templates with the Form Wizard and add them to this list. For more information about adding custom templates, see the section "Creating a Form Template" on page 366.

5 **(Optional) Select a folder for storing the form.**

If you select a folder name, the form will be placed in this folder. Folders can be used to organize the objects of your application and are managed on the Home page of the Explorer. For more information, refer to the "Home Page" paragraph on page 103. By default, the form is created at the Top Level, i.e., outside any storage folder.

6 **Select the fields you want on your form.**

For complete information about selecting fields for the form, refer to the section "Creating a New Form using the Wizard" on page 350.



7 **If you want to build the new form and modify it in the Form editor, click Edit.**
OR

If you want to build the form and set it as the current form, click Use.
OR
If you want to customize the new form with the Form Wizard's advanced options, click Advanced...

Using the Form Wizard's Advanced Options

When you create a List form, the Advanced screen of the Form Wizard has the Fields, Options and (for display forms only) Buttons pages.

Fields Page

This page is similar to the Fields page for Detail forms, except that grouping fields is not relevant for List forms and is not available. For more information, refer to the "Fields Page" paragraph on page 356.

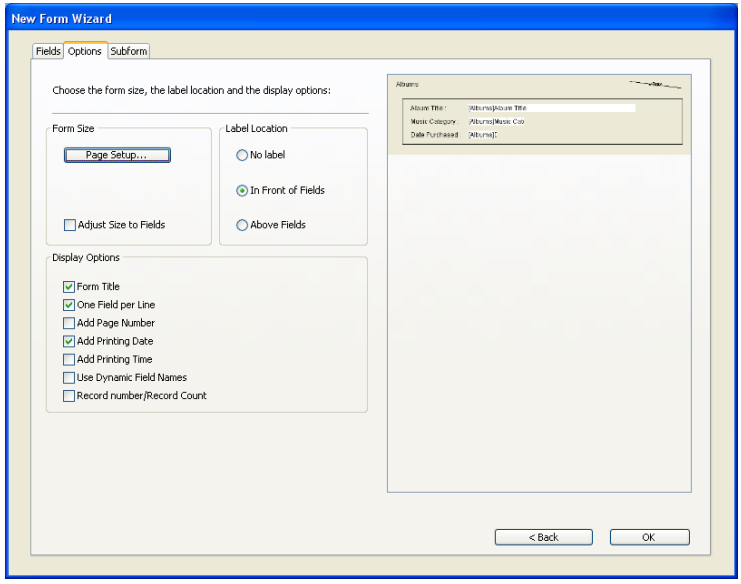
Options Page

This page is similar to the Options page for Detail forms, with the following differences:

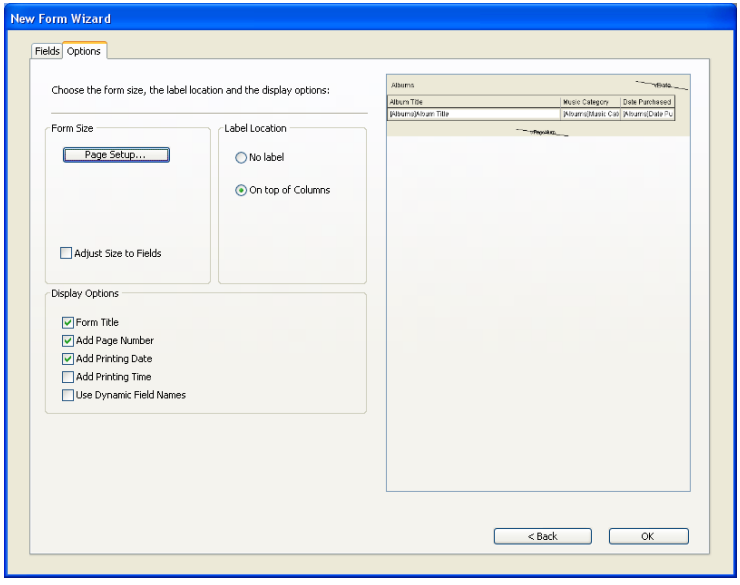
- **For Printing forms** (Detail Form for Printing and List Form for Printing):
  - The "Form Size" area includes a **Page Setup...** button. This button displays the current Print Setup dialog box, which can be used to choose the page size for the printed report. 4D adjusts the form size and the preview area according to the page format that you have chosen.

- In the “Display Options” area, you can select the variables that you want to insert into your reports; you can display the form title, page number, date and time of printing:

Detail Form for Printing



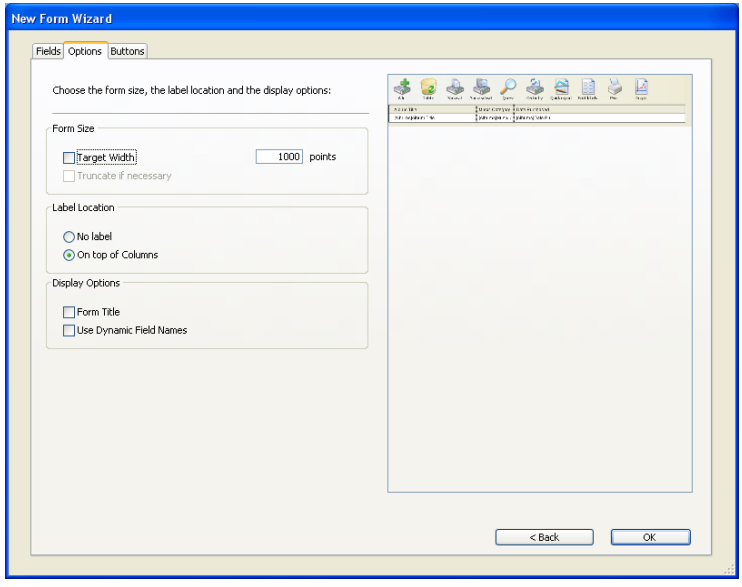
List Form for Printing



- For the **List form** type: the “Form Size” area also includes an additional **Target Width** option.

If you do not check this option, the output form width will be calculated automatically with respect to the width of the form fields. If you click **Target Width** and enter a target width in pixels, the Form Wizard will try to fit all the fields on the form by reducing the widths of the fields. If you also click **Truncate if necessary**, the Form Wizard will then remove one or more fields to make the form width less than or equal to the target width.
If you don't check **Truncate if necessary**, the form's width may be slightly larger than the target width.

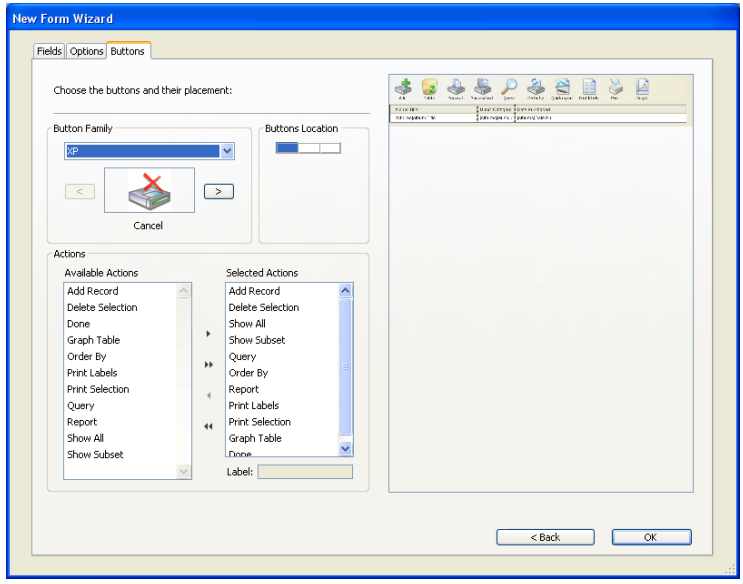
List Form



Buttons Page

This page is similar to the Buttons page for Detail forms, except that different buttons are available. The Buttons page is not available for the List and Detail forms for Printing form types.

List Form



The buttons are not automatic in the same sense as buttons for input forms; instead, 4D automatically creates an object method for certain buttons. Using the Form editor, you can edit the object methods.

For more information on working with the Buttons page, see the "Buttons Page" paragraph on page 359.

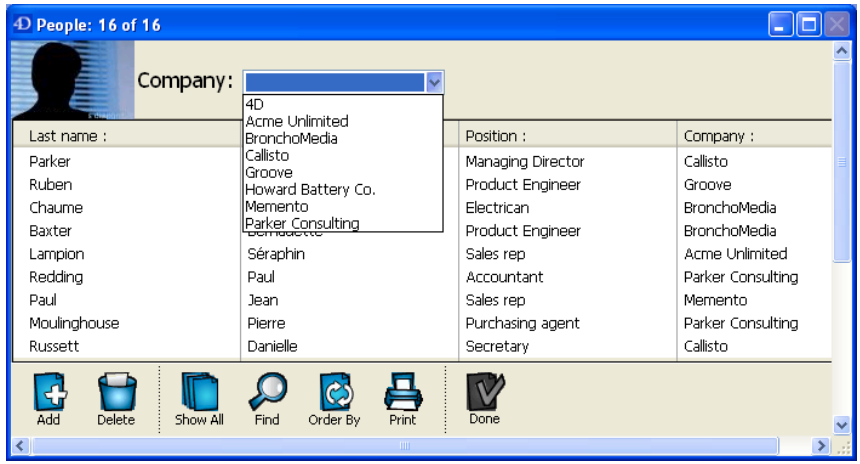
Modifying an Output Form in the Form Editor

A List form that is created using the Form Wizard works well for displaying a list of records. If needed, you can make the following simple modifications:

- Replace fields with variables and add methods,
- Use the platform interface, appearance, font attributes, fill, border, or color options to modify individual objects on the form,
- Change the widths of the fields or variables,
- Move the output control lines,
- Add a custom graphic in the Header area,

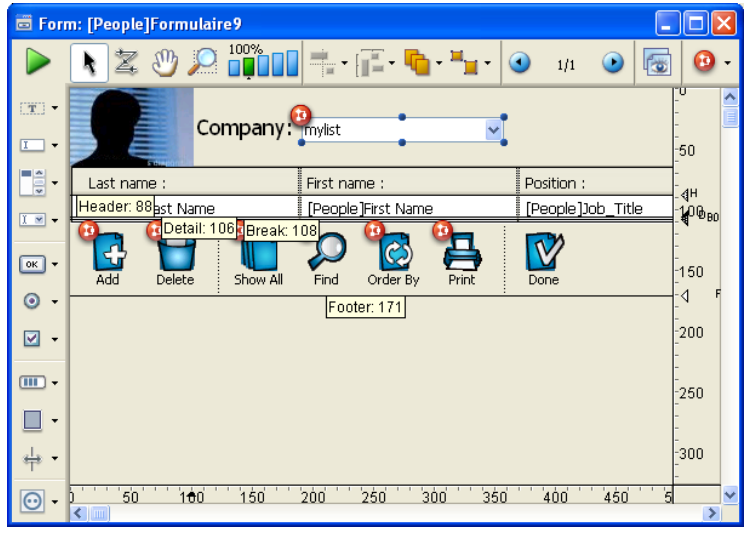
- Add variables in the Header or Footer area.

The following is a custom output form being used for managing personnel:



The control buttons have been placed in the footer area and a custom picture has been placed in the header area.
The Header area also includes a drop-down list that lets you select the company whose employees you want to display.

Here is what the design for this form looks like in the Form editor:



The control lines can be moved, for example, in order to adapt to the size of the picture inserted into the header area.

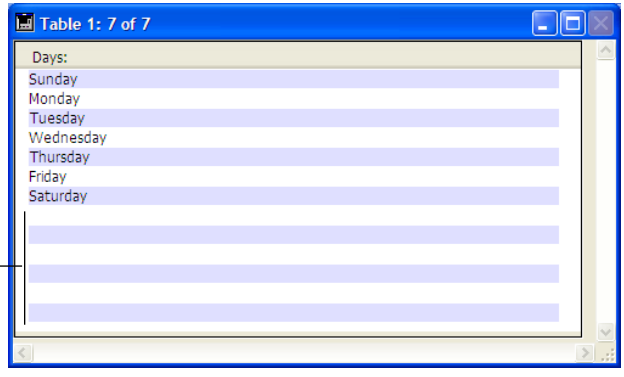
Displaying Several Lines Per Record

You may want to display fields on more than one line. 4D allows you to use several lines for each record. Expand the Detail area so that more than one line appears in this area. When you use several lines for a single record, graphic elements such as lines and boxes can be useful to separate fields and records.

Here is an example of an output form for a phone message management database which uses two lines per record:

Managing Empty Lines

In list mode, 4D displays, if necessary, empty lines below the last displayed records, so that it fills the entire window:



When the window is displayed, the On Display Detail form event is generated for each empty line of the list. In this case, there is no current record: Record number returns -1 and Selection number does not return a significant value. The Displayed line number command lets you find out the number of the line currently being displayed.

System Highlight Color

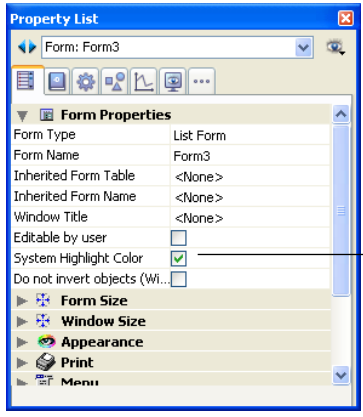
In versions of 4D prior to 2004, highlighted records in lists were shown using “reverse video”: the color of each pixel in the selection is reversed (for example, white becomes black, red becomes green and vice versa). This principle included the background and text of objects:

Last Name	First Name	Company
Durand	Pierre	Omega
Dupond	Paul	Omega
Martin	Jean	Alpha
Rebel	Eleonor	Alpha
Gillot	Carl	Curling Braces
Lemarchand	Guy	Curling Braces

Beginning with version 2004, for better interface conformity, 4D uses the operating system highlight settings in new forms:

Last Name	First Name	Company
Durand	Pierre	Omega
Dupond	Paul	Omega
Martin	Jean	Alpha
Rebel	Eleonor	Alpha
Gillot	Carl	Curling Braces
Lemarchand	Guy	Curling Braces

It is possible to take advantage of this new interface in converted databases using the **System Highlight Color** option. This option is only available for output forms in converted databases:

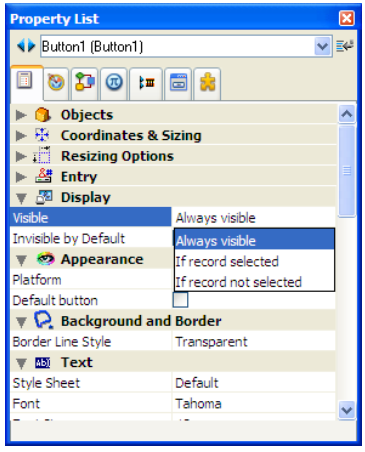


When this option is checked, 4D disables the reverse video and uses the system highlight color to represent selected items. Be careful, for a depiction in compliance with interface standards, this option must be accompanied by appropriate text and background color settings for objects located within the form body area:

- **Automatic** option for the text color of each object,
- **Transparent** option for the background color of each object,
- **Automatic** option for the color of the rectangle located beneath the form body area.

Object Visibility

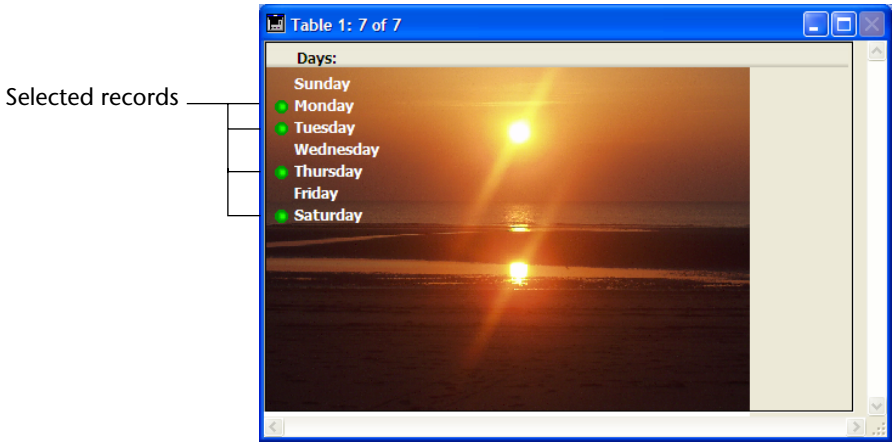
The **Visible** display property is available for all objects in the Form editor when the form is a list form.



This property accepts three values:

- Always visible
- If record selected
- If record not selected.

This property is only used when drawing objects located in the body of a list form. It tells 4D whether or not to draw the object depending on whether the record being processed is selected/not selected. This property allows you to represent a selection of records using visual attributes other than colors:



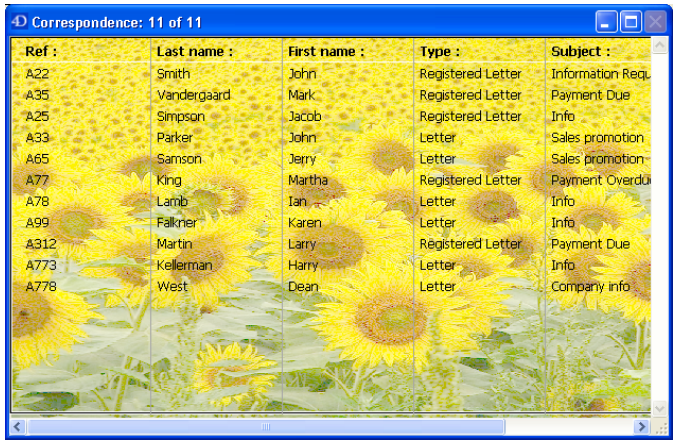
4D does not take this property into account if the object was hidden using the SET VISIBLE command; in this case, the object remains invisible regardless of whether or not the record is selected.

If empty lines are displayed after the last records (please see the "Managing Empty Lines" paragraph on page 654), 4D draws the objects with the **Always visible** or **If record not selected** property in each empty line.

Management of Background Pictures

Output forms include a specific mechanism allowing the display of background pictures: for this to work, the picture must begin in the header area and overlap into the break area. This way it will be drawn only once, by extending its height:

Form editor



Form in use

This mechanism takes the picture display properties (truncated, scaled to fit, replicated) into account.

Warning: This mechanism does not work with active objects. If an active object (field or variable) located in the detail of the form overlaps into the header or footer area, it is not displayed.

Modifying Output Forms for Reports

In 4D, you can produce a report using either a form or using the Quick Report editor.

There are several advantages to using a form for a report: you can use graphic elements on the form, you can control the placement of report elements precisely, you can use methods to perform calculations, and you can use headers to identify each subsection of a break report. Form reports can also be customized by each user with the help of the user form editor (see the chapter “User Forms” on page 685).

You can use a form to create reports that:

- Require a non-columnar format,

- Display subforms,
- Contain embedded graphics,
- Require special graphic elements, such as hairlines.

Printed reports, unlike screen display forms, can make use of the Break area at the end of the report. For material that appears at the bottom of each page, printed reports use the Footer area.

This section provides suggestions and directions specifically for designing forms to be used for printing. It covers lists, reports that print one record per page, expandable areas for printing invoices and variable text fields, form letters, and custom mailing labels.

Printed Columnar Reports

Lists that display columns of information are common in printed reports. You might publish lists of telephone numbers, prices, results, specifications, or parts.

When you create a columnar report using the Form Wizard, you choose **List Form for Printing** as the Form Type.

Like the lists you design to display records on the screen, a printed list presents columns of information, can include column headings above each column, and can use graphic elements to enhance or clarify the report.

The Break area, which is printed once at the end of the report, is used for calculating totals. For a discussion of using methods for calculating totals, see the section "An Example Report" on page 675.

One Record Per Page Reports

You may need to print one record per page. For example, you may want to use an invoicing database to print a copy of each invoice for your records.

When you need to create such a report, you choose **Detail Form for Printing** as the Form Type in the Form Wizard.

Place the Header (H) control line at the top of the page and arrange the fields and other report elements below it.

If your form displays records in a subform, be sure that the subform is set to print with a fixed frame so that the records do not wrap onto additional pages. For information about printing with a fixed frame, see the section "Printing Subforms, Pictures, and Text Fields" on page 677.

Drag the Detail (D), Break (B0), and Footer (F) control lines to the bottom of the page to ensure that only one record is printed per page.

Using Subforms

You are probably very familiar with an invoice. A typical invoice shows a name and billing address, a shipping address, a series of items or services purchased, and a total.

An invoicing database includes an output form for printing full-page invoices. For example, an invoice draws information from two tables: an [Orders] table that provides the customer information (bill to and ship to addresses) and a [Line Items] table that provides the line items. The total for the order is calculated and kept in the [Orders] table.

The form for an invoice is created in the [Orders] table and uses a subform area for the line items. The subform area can expand during printing to print all the line items, even if the invoice requires a second page. For more information, see the section "Printing Subforms, Pictures, and Text Fields" on page 677.

The Detail area is expanded to the full-page size. One invoice is printed for each sales order, but as many line items as necessary are printed in the line items area of the invoice.

Report with a Text Field

Many databases allow the user to enter notes or descriptions in a text field. That same text field can be printed as an expandable area in the Detail area.

Text fields can be expanded during printing to accommodate all of the data. For more information about printing text fields, see the section "Printing Subforms, Pictures, and Text Fields" on page 677.

Custom Mailing Labels

If you want to create special mailing labels, you can design a custom output form for them. The design can use graphic elements, any available fonts, and variables.

Tip It is often quicker and easier to create mailing labels using the Label editor. For more information about the Label editor, refer to the chapter “Labels” on page 1031.

The creation of output forms used for printing mailing labels is detailed in the “Printing Labels” paragraph on page 680.

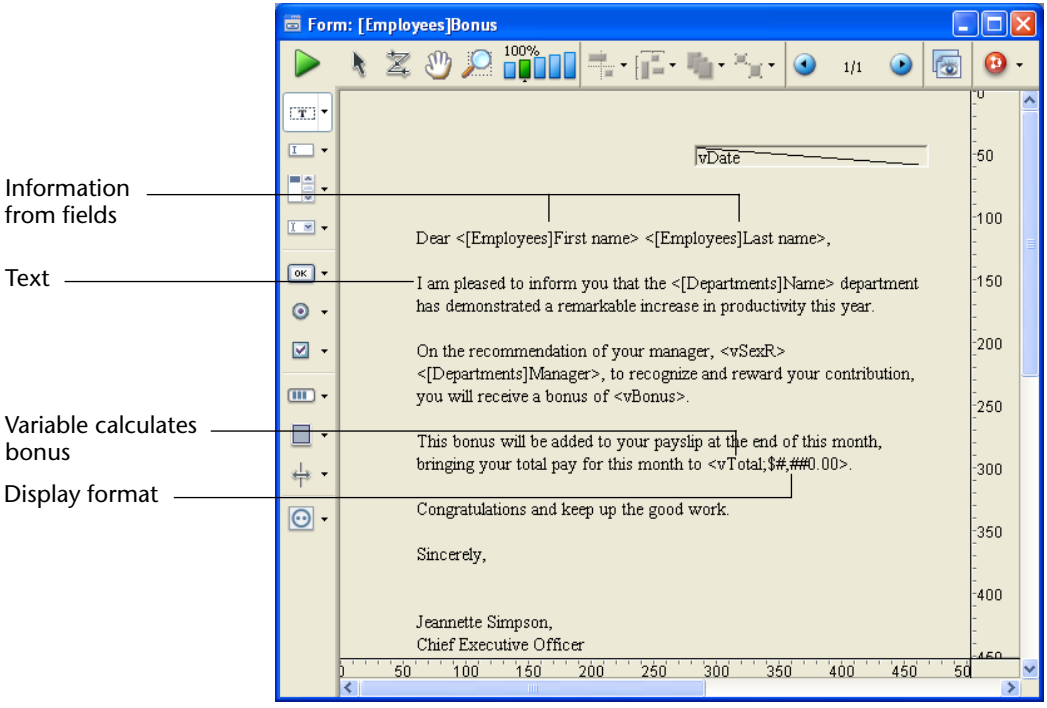
Creating Mail-Merge Documents

You can handle mail-merge tasks using an output form that embeds fields, field or table labels, or variables in a static text area. You can create documents that are the same in every respect except for names, addresses, and any special calculations you want to perform.

Mail-merge documents may be useful when you want to announce a special offer or inform your customers or partners of a business development (such as a move to a new location or a significant personnel change). Another typical use of mail-merge is to inform customers that their account is due. You can create a variable and a method to calculate the exact amount.

When you create a form that does mail-merge, choose the **Detail Form for Printing** option in the Form Type drop-down list of the Form Wizard. In the Form editor, create a text area that will contain both the static text and the fields, variables, or table or field labels that will change for each record. You then embed fields or variables in the text area. During printing, values from the fields or variables are inserted in the text.

The figure below shows fields and variables placed in a text area.



► To create an output form for a mail-merge:

1 **In the Form Wizard, create a new form using the Detail Form for Printing option in the Form Type drop-down list.**

2 **Click Edit to open the form in the Form editor.**

3 **Select the Text Area tool and create a text area in the Detail area.**

4 **Type the text you want in your form letter, placing field and variable names between less than (<) and greater than (>) symbols where you want information from fields or variables to be inserted.**

You can use a field from any table in the database. Fields from the master table do not have to specify the table name; they can be entered like this: <*field*>. Fields from other tables must specify the table name; they are entered like this: <*[tablename] field*>. When the form is printed, the information from the field for each record replaces the <*field*> element in the text area.

To insert table labels, enter: *<?[Table Name]>* or *<?[N]>* where N is the creation order for the table. To insert field labels, enter: *<?[Table Name]Field Name>* or *<?[X]Y>* where *X* is the creation order for the table and *Y* is the creation order for the field or *<?Y>* to insert a field of the current table. For more information, refer to "Inserting Dynamic Table and Field Names" on page 487.

A variable must be assigned a value in an object or form method.

You can specify how an embedded field or variable is displayed by inserting a semicolon followed by a display format directly after the field or variable name. For example, the mail-merge document shown above includes a display format for the vTotal variable. The embedded variable <vTotal;$###,##0.00> calculates the amount of the person's total pay for the month and displays it in a dollar format. For more information about display formats, see the section "Display Formats" on page 521.

4D provides a shortcut for inserting fields in a text area. The shortcut allows you to choose the field name from a drop-down menu.

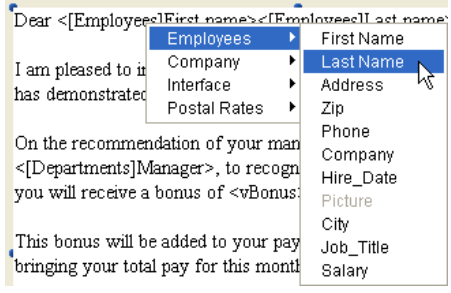
► To insert a field using the shortcut:

1 **Click with the text tool to place the insertion point where you want to insert the field.**

2 **Position the pointer in the text area and hold down the Alt key (Windows) or Option key (Mac OS) while you press and hold down the mouse button.**

4D displays a pop-up menu of fields from the master table from which you can choose the field you want.

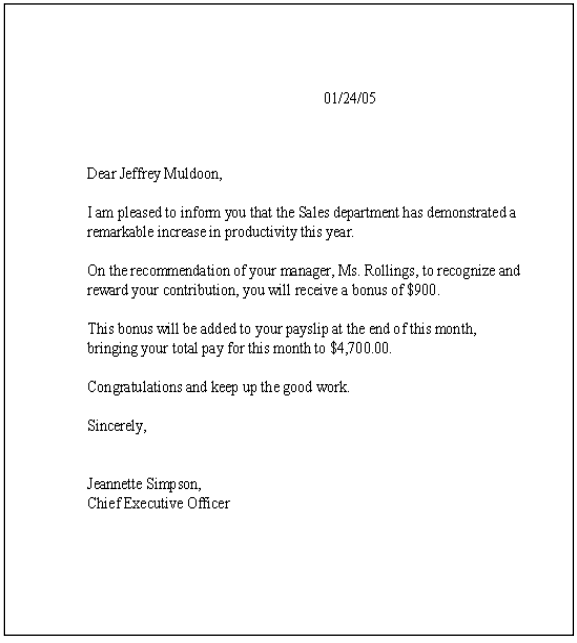
To choose a field from another table in the database, hold down Shift+Alt (Windows) or Shift-Command (Mac OS) while you press and hold down the mouse button.



4D displays a hierarchical menu of tables and fields in the database. You can choose the table and field you want.

4D places <*field*> or <*[table]field*> in the text area at the insertion point.

When the report is printed, the values of the fields and variables embedded in the letter appear for each record.



01/24/05

Dear Jeffrey Muldoon,

I am pleased to inform you that the Sales department has demonstrated a remarkable increase in productivity this year.

On the recommendation of your manager, Ms. Rollings, to recognize and reward your contribution, you will receive a bonus of $900.

This bonus will be added to your payslip at the end of this month, bringing your total pay for this month to $4,700.00.

Congratulations and keep up the good work.

Sincerely,

Jeannette Simpson,
Chief Executive Officer

01/24/05

Dear Alan Hull,

I am pleased to inform you that the Marketing department has demonstrated a remarkable increase in productivity this year.

On the recommendation of your manager, Mr. Trump, to recognize and reward your contribution, you will receive a bonus of $800.

This bonus will be added to your payslip at the end of this month, bringing your total pay for this month to $4,980.00.

Congratulations and keep up the good work.

Sincerely,

Jeannette Simpson,
Chief Executive Officer

Basic Steps for Creating a Printed Report

You can perform most or all of these steps depending on the type of the report you want to print. The actual order of steps is not critical. Typically, you design a report, preview it on screen, and then return to the form to make adjustments. The order given here is typical:

1 **Create a form using the Form Wizard.**

For information about creating a form, refer to Chapter 6.

2 **Move the control lines so that you will have enough space to place the various elements of the report.**

For information about moving control lines, see the section "Moving Output Control Lines" on page 645.

3 **Move elements into position.**

You can place fields, text, non-enterable objects, and graphic objects. Take time to align the objects to one another and to check the justification of data to be displayed in fields and active objects.

4D displays page size guidelines in the form so that you can place elements for printing.

For information about placing fields and active objects in a form, refer to Chapter 7.

4 **Adjust the control lines if necessary.**

Working with the relationship between control lines and elements to be printed, you can create the right amount of space for printing the Header, Detail, Break, and Footer elements.

5 **If necessary, create methods to calculate values, concatenate strings of characters, print additional text, and display the date, time, and page numbers.**

For information about creating methods, refer to "Using Object Methods with Fields and Objects" on page 616. For complete information about using object methods, refer to the *4D Language Reference* manual.

For information about calculating subtotals and other summary calculations (averages, totals, etc.), refer to "Reports with Breaks" on page 666.

Be sure that you activate the appropriate events in the Events section of the object properties for each method you use; otherwise the method will not execute during printing.

6 **Click on the execute button of the form to test the report.**

7 **Create the selection of records you want to use for the report.**

For information about creating a selection of records, refer to the "Selecting and Searching Records" paragraph on page 950.

8 **Sort the records according to how you have designed the report.**

For information about sorting records, refer to the "Sorting Records" paragraph on page 987.

9 **Preview the report by printing it to the screen.**

The report form uses its own print settings. If you make any changes to the Page Setup dialog box, be sure to make the same changes to the form print settings in the Design environment (see the "Form Print Settings" paragraph on page 480).

If the report form needs to be adjusted, return to the Design environment to make any necessary changes.

10 **Print the report.**

For information about printing, refer to the "Printing Records" paragraph on page 994.

Reports with Breaks

4D can print reports that work with Breaks and Break Headers. A Break is created when you sort the records.

Suppose you have a collection of compact discs that you keep track of in a 4D database and you want to print a list that arranges the information by artist. When you sort the records by artist, all the records fall into distinct groups. The "Break" occurs after the last record in each group is printed.

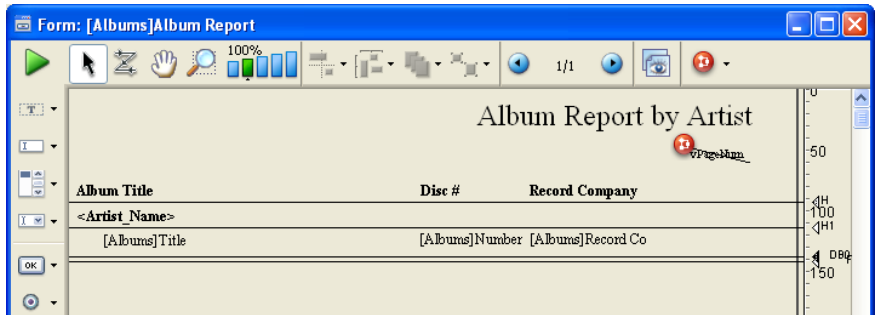
Here is how the report looks when previewed on screen:

Album Report by Artist

Page 4

Album Title	Disc #	Record Company
THE PLANETS	023	Deutsche Grammophon
THE PLANETS, SUITE DE BALLET, OP. 10	250	Enigma Classics
Harry Belafonte		
ALL TIME GREATEST HITS VOL.1	025	BMG Music
PARADISE IN GAZANKULU	028	Capitol Records
Hector Berlioz		
SYMPHONIE FANTASTIQUE	192	Enigma Records
Hiroshima		
ANOTHER PLACE	036	
Huey Lewis And The News		
FORE!	031	Chrysalis Records
INXS		
KICK	203	Atlantic Recording Corporation
Jean-Luc Ponty		
COSMIC MESSENGER	029	Atlantic Recording Corporation
Jerry Goodman		
ARIEL	073	Private Music, Inc.
Joaquin Rodrigo		
CONCIERTO DE ARANJUEZ	233	Enigma Classics
Joe Sample		
SPELLBOUND	179	Warner Bros. Records Inc.
Johann Sebastian Bach		
BACHBUSTERS	080	TELARC DIGITAL
BRANDENBURG CONCERTOS NOS. 1, 2, & 3	125	Enigma Records
BRANDENBURG CONCERTOS NOS. 4, 5 & 6	163	Enigma Records
ORGAN FAVOURITES	248	Enigma Classics
Johann Strauss, Jr.		
STRAUSS FESTIVAL VOLUME 1: FAMOUS WALTZES, PC	096	Enigma Records
STRAUSS FESTIVAL VOLUME 2	159	Enigma Records
Johannes Brahms		
HUNGARIAN DANCES NOS. 1-21	243	Enigma Classics
HUNGARIAN DANCES NOS. 1-21	146	Enigma Records
SYMPHONY NO.4 ACADEMIC FESTIVAL OVERTURE	127	Enigma Records
Joni Mitchell		
COURT AND SPARK	108	Asylum Records
Joseph Haydn		

4D provides features that you can use to display the information attractively. Here is this form in the Design environment:



In order to build a report that uses Break levels and Headers, you must first initiate Break processing. For more information about the methods you can use to initiate Break processing, refer to "Initiating Break Processing" on page 670.

A Break Header is printed once before the group of records it refers to and a Break is printed once after the group of records it refers to. In the illustration on the previous page, the Break is called a "level 1 Break" and the Break Header is called a "level 1 Break Header," because the Break occurs as a result of the first field sorted.

You can use up to nine break levels. If you use Subtotal to initiate Break processing, you need to sort on one more field than the number of Breaks you use. In this case, if you use one Break level, you must sort on two fields. If you use three Break levels, you must sort on four fields, and so on.

This section explains how to create reports using Breaks. A full-scale example is provided at the end of this section. If you have any trouble understanding anything in this section, try reading the whole section and then coming back to the trouble spot. The concepts are interrelated and are often easier to understand in context than by themselves.

Creating Additional Control Lines

The report examples shown in this section use Break levels and Break Headers. To create areas for these features, you create additional control lines.

The Form editor always starts with the original control lines, labeled H, D, B0, and F. B0 stands for "Break at level 0." Level zero takes in all the records; it occurs after all the records are printed. Additional Break control lines are designated with numbers. A control line labeled B1 stands for "Break at level 1." A level one Break occurs after the records grouped by the first sorted field are printed.

Label	Explanation	Prints after groups created by:
B1	Break at level 1	First sorted field
B2	Break at level 2	Second sorted field
B3	Break at level 3	Third sorted field

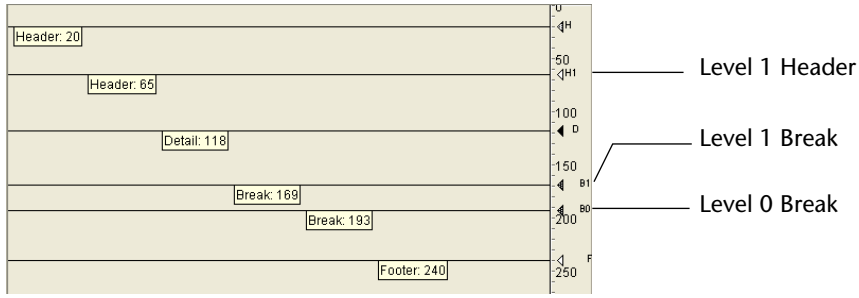
H stands for "Header," which is printed at the top of each page. Additional Header control lines are associated with Breaks. H1 stands for "Header at level 1." A level 1 Header is printed just before the records grouped by the first sorted field are printed.

Label	Explanation	Prints before groups created by:
H1	Header at level 1	First sorted field
H2	Header at level 2	Second sorted field
H3	Header at level 3	Third sorted field

You create additional control lines by holding down the **Alt** key (Windows) or **Option** key (Mac OS) while clicking the appropriate control marker. You use a Break control line to create a Break area for the designated level. You use a Break Header control line to create a Break Header area at the designated level.

If you use the Subtotal function to initiate Break processing, you should create a Break area for every level of Break that will be generated by the sort order, minus one. If you do not need anything printed in one of the Break areas, you can reduce its size to nothing by placing its marker on top of another control line. If you have more sort levels than Break areas, the last Break area will be repeated during printing.

The figure below shows additional control lines.



► To create new Break or Break Header control lines:

1 **Hold down the Alt key (Windows) or Option key (Mac OS) and click any Break (B) or Break Header (H) marker.**

4D creates a new Break or Break Header control line.

The new line is positioned behind the existing control line; to see the new control line, you need to drag the existing line off of it.

2 **Drag the existing line away from the new line.**

3 **Position both lines where you want them.**

► To delete Break or Break Header control lines that you have created:

1 **Hold down the Ctrl key (Windows) or Command key (Mac OS).**

2 **Click on the Break, Break Header, or label of the control line that you want to delete.**

4D deletes the control line and, if necessary, renumbers the remaining lines.

You cannot delete the original control lines H, D, B0, and F.

Initiating Break Processing

To allow 4D to print Break Header areas, accumulate subtotals, and perform other aspects of Break processing, you must first initiate Break processing in the report form. You initiate Break processing by either:

- Placing the Subtotal function in an object or form method,
- Executing the BREAK LEVEL and ACCUMULATE commands before printing the report.

If you use the Subtotal function, you must also sort the records on one more sort field than the number of Break levels you use. For example, if you use two levels of Breaks in your report, you must sort on three fields.

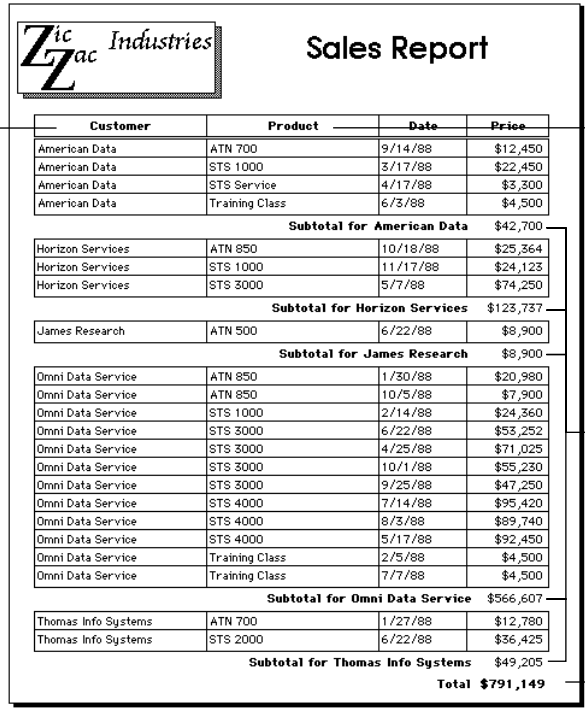
For more information about initiating Break processing, refer to the *4D Language Reference* manual.

Reports with Subtotals

This section describes in detail how the sort order affects reports and how to use additional Break areas for creating subtotals.

4D can automatically calculate and print totals and subtotals.

The figure below shows a report that calculates subtotals for each customer and a total at the end of the report.



Zic Zac Industries

Sales Report

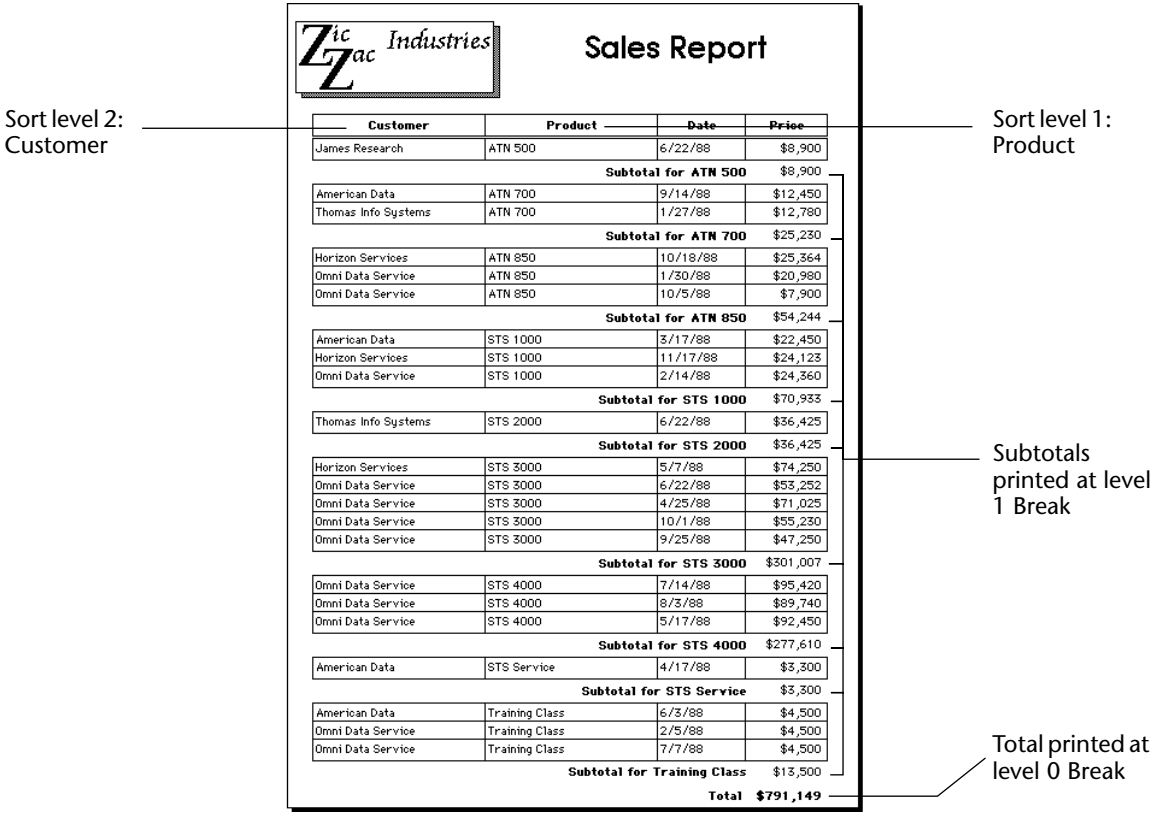
Customer	Product	Date	Price
American Data	ATN 700	9/14/88	$12,450
American Data	STS 1000	3/17/88	$22,450
American Data	STS Service	4/17/88	$3,300
American Data	Training Class	6/3/88	$4,500
	Subtotal for American Data		$42,700
Horizon Services	ATN 850	10/18/88	$25,364
Horizon Services	STS 1000	11/17/88	$24,123
Horizon Services	STS 3000	5/7/88	$74,250
	Subtotal for Horizon Services		$123,737
James Research	ATN 500	6/22/88	$8,900
	Subtotal for James Research		$8,900
Omni Data Service	ATN 850	1/30/88	$20,980
Omni Data Service	ATN 850	10/5/88	$7,900
Omni Data Service	STS 1000	2/14/88	$24,360
Omni Data Service	STS 3000	6/22/88	$53,252
Omni Data Service	STS 3000	4/25/88	$71,025
Omni Data Service	STS 3000	10/1/88	$55,230
Omni Data Service	STS 3000	9/25/88	$47,250
Omni Data Service	STS 4000	7/14/88	$95,420
Omni Data Service	STS 4000	8/3/88	$89,740
Omni Data Service	STS 4000	5/17/88	$92,450
Omni Data Service	Training Class	2/5/88	$4,500
Omni Data Service	Training Class	7/7/88	$4,500
	Subtotal for Omni Data Service		$566,607
Thomas Info Systems	ATN 700	1/27/88	$12,780
Thomas Info Systems	STS 2000	6/22/88	$36,425
	Subtotal for Thomas Info Systems		$49,205
	Total		**$791,149**

These records have been sorted by customer and product. After the records for each customer have been printed, the subtotal for the customer is calculated and printed. After the records for the entire report have been printed, the total is calculated and printed.

As you can see in the figure above, the subtotals are calculated and printed after the records for each customer. 4D knows when to perform the calculation and print the subtotal because it has been instructed to do so when the value in the first sorted field changes (where it "breaks").

The customer subtotal is calculated in what is called a level 1 Break because it is based on a change in the first sorted level (in this case, the Customer field). The grand total is calculated in a level 0 Break. A level 0 Break includes all the records and occurs at the end of the report.

The figure below shows another example of calculations during a Break, using the same records, but with a different sort order to create a different Break. This time the records have been sorted by product and customer. The subtotals are calculated when the value in the Product field changes. This is still a level 1 Break, but the Break is on a different field.



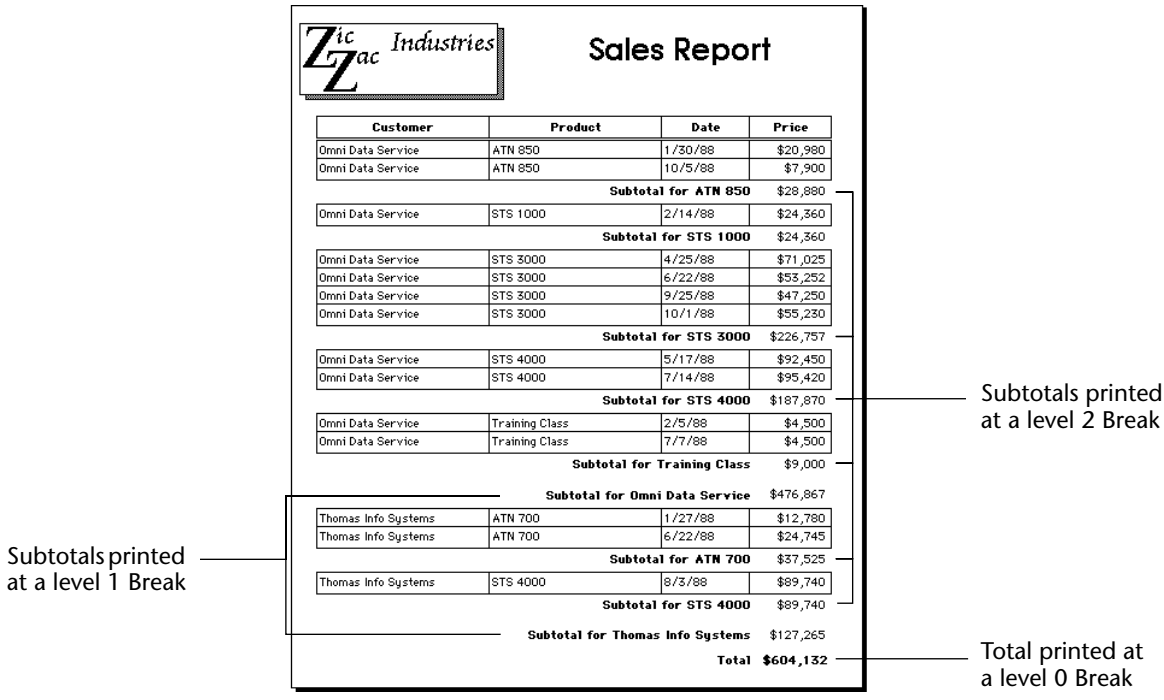
Customer	Product	Date	Price
James Research	ATN 500	6/22/88	$8,900
	Subtotal for ATN 500		$8,900
American Data	ATN 700	9/14/88	$12,450
Thomas Info Systems	ATN 700	1/27/88	$12,780
	Subtotal for ATN 700		$25,230
Horizon Services	ATN 850	10/18/88	$25,364
Omni Data Service	ATN 850	1/30/88	$20,980
Omni Data Service	ATN 850	10/5/88	$7,900
	Subtotal for ATN 850		$54,244
American Data	STS 1000	3/17/88	$22,450
Horizon Services	STS 1000	11/17/88	$24,123
Omni Data Service	STS 1000	2/14/88	$24,360
	Subtotal for STS 1000		$70,933
Thomas Info Systems	STS 2000	6/22/88	$36,425
	Subtotal for STS 2000		$36,425
Horizon Services	STS 3000	5/7/88	$74,250
Omni Data Service	STS 3000	6/22/88	$53,252
Omni Data Service	STS 3000	4/25/88	$71,025
Omni Data Service	STS 3000	10/1/88	$55,230
Omni Data Service	STS 3000	9/25/88	$47,250
	Subtotal for STS 3000		$301,007
Omni Data Service	STS 4000	7/14/88	$95,420
Omni Data Service	STS 4000	8/3/88	$89,740
Omni Data Service	STS 4000	5/17/88	$92,450
	Subtotal for STS 4000		$277,610
American Data	STS Service	4/17/88	$3,300
	Subtotal for STS Service		$3,300
American Data	Training Class	6/3/88	$4,500
Omni Data Service	Training Class	2/5/88	$4,500
Omni Data Service	Training Class	7/7/88	$4,500
	Subtotal for Training Class		$13,500
	Total		**$791,149**

Additional Break Levels

You can provide additional summary calculations by adding another sort level and another Break level.

The following figure shows sales records sorted by customer, product, and salesperson. Summary calculations show two sets of subtotals: one subtotal for each customer, and, within each customer, subtotals for each product for the customer. Finally, this report calculates a total for the entire company.

These are examples of calculations performed at level 2 Breaks, at level 1 Breaks, and at the level 0 Break.

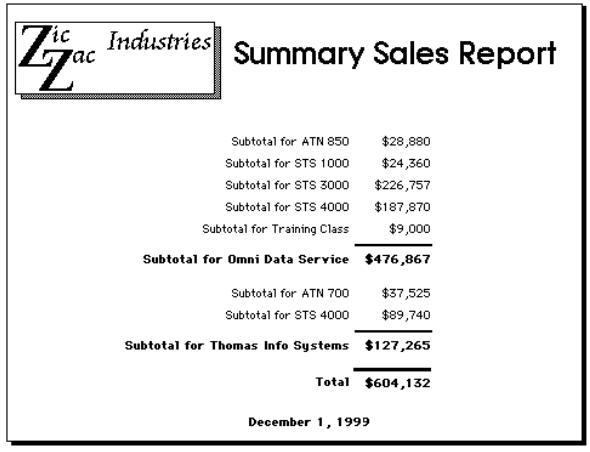


Customer	Product	Date	Price
Omni Data Service	ATN 850	1/30/88	$20,980
Omni Data Service	ATN 850	10/5/88	$7,900
		Subtotal for ATN 850	$28,880
Omni Data Service	STS 1000	2/14/88	$24,360
		Subtotal for STS 1000	$24,360
Omni Data Service	STS 3000	4/25/88	$71,025
Omni Data Service	STS 3000	6/22/88	$53,252
Omni Data Service	STS 3000	9/25/88	$47,250
Omni Data Service	STS 3000	10/1/88	$55,230
		Subtotal for STS 3000	$226,757
Omni Data Service	STS 4000	5/17/88	$92,450
Omni Data Service	STS 4000	7/14/88	$95,420
		Subtotal for STS 4000	$187,870
Omni Data Service	Training Class	2/5/88	$4,500
Omni Data Service	Training Class	7/7/88	$4,500
		Subtotal for Training Class	$9,000
		Subtotal for Omni Data Service	$476,867
Thomas Info Systems	ATN 700	1/27/88	$12,780
Thomas Info Systems	ATN 700	6/22/88	$24,745
		Subtotal for ATN 700	$37,525
Thomas Info Systems	STS 4000	8/3/88	$89,740
		Subtotal for STS 4000	$89,740
		Subtotal for Thomas Info Systems	$127,265
		Total	**$604,132**

The subtotal calculations are performed only for the group of records that precedes the Break. For example, a subtotal is calculated for each product sold to each customer. The subtotal for the customer is calculated for all products sold to that customer.

Summary Reports

You can create a report that prints only summary information. Such a report displays only the subtotals and totals with appropriate additional text. The following illustration shows a report with only summary information.



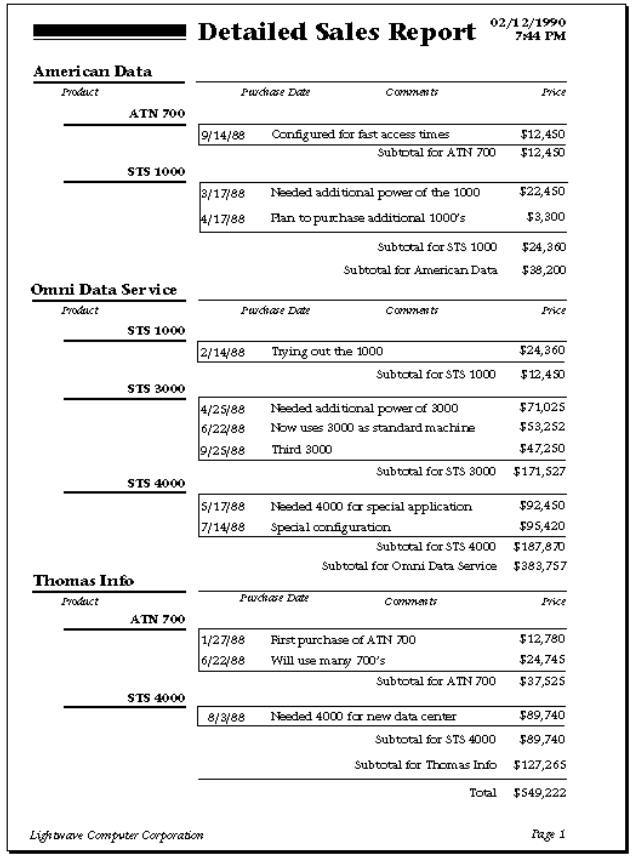
In this report, the records are sorted by customer, product, and date and the calculations are performed during the Breaks created by the sort order. The Detail area for each record is not printed; the records are used only to provide values for the calculations. Notice that 4D goes through the records from the first to the last during the printing of the report so that it can calculate these figures. (You create this kind of report by placing the Detail control line on top of the Header control line, leaving no space for details to print.)

You can ask 4D to perform additional calculations on a list including subtotals, averages, maximum and minimum values, page totals to be printed in a footer, and weighted averages. These calculations, and others, use 4D's summary functions. For information about how to use summary functions, refer to the *4D Language Reference* manual.

An Example Report

This section describes an example report and shows how the finished report is related to a form in the Form editor and to the methods that control the printing.

The following illustration shows a finished report.



Detailed Sales Report 02/12/1990 7:44 PM

American Data

Product	Purchase Date	Comments	Price
ATN 700			
	9/14/88	Configured for fast access times	$12,450
		Subtotal for ATN 700	$12,450
STS 1000			
	3/17/88	Needed additional power of the 1000	$22,450
	4/17/88	Plan to purchase additional 1000's	$3,300
		Subtotal for STS 1000	$24,360
		Subtotal for American Data	$38,200

Omni Data Service

Product	Purchase Date	Comments	Price
STS 1000			
	2/14/88	Trying out the 1000	$24,360
		Subtotal for STS 1000	$12,450
STS 3000			
	4/25/88	Needed additional power of 3000	$71,025
	6/22/88	Now uses 3000 as standard machine	$53,252
	9/25/88	Third 3000	$47,250
		Subtotal for STS 3000	$171,527
STS 4000			
	5/17/88	Needed 4000 for special application	$92,450
	7/14/88	Special configuration	$95,420
		Subtotal for STS 4000	$187,870
		Subtotal for Omni Data Service	$383,757

Thomas Info

Product	Purchase Date	Comments	Price
ATN 700			
	1/27/88	First purchase of ATN 700	$12,780
	6/22/88	Will use many 700's	$24,745
		Subtotal for ATN 700	$37,525
STS 4000			
	8/3/88	Needed 4000 for new data center	$89,740
		Subtotal for STS 4000	$89,740
		Subtotal for Thomas Info	$127,265
		Total	$549,222

Lightwave Computer Corporation *Page 1*

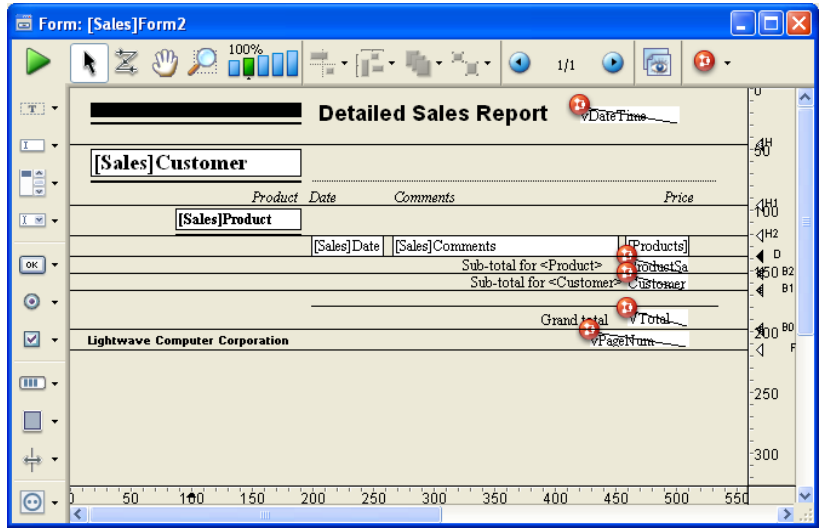
In this example, the page Header contains the date, the time, and the report title. The Break Headers contain the customer name and column headings for the information presented during the first Break. The Detail areas contain data drawn directly from the records. The level 2 Break areas contain subtotals for products for each customer. The level 1 Break areas contain subtotals for each customer and the level 0 Break area contains a total for the report. The Footer contains the page number.

The report is sorted on one more level than Break levels. In this report, the sorted fields are Customer, Product, and Date.

4D requires one more sort level than Break level for the Break processing method used in this report.

The Report Form

The following illustration shows the report form that created the report shown on the previous page.



Each control line in the form defines the bottom of its area. Whatever is placed in the form is printed at the appropriate place in the report. The Header area contains the elements that will be printed at the top of each page, the Detail area contains the elements that will be printed for each record, and so on. The following table shows what each of these control lines means.

Label	Explanation	Effect
H	Header area	Printed once at the top of each page
H1	L1 Header area	Printed once before each level 1 Break
H2	L2 Header area	Printed once before each level 2 Break
D	Detail area	Printed once for each record in the selection
B2	L2 Break area	Printed once at each level 2 Break (when the value in the second Sorted field changes)
B1	L1 Break area	Printed once at each level 1 Break (when the value in the first Sorted field changes)
B0	L0 Break area	Printed once at the end of the report
F	Footer area	Printed once at the bottom of each page

You can have additional Break areas and additional Break Header areas.

The Report Object Methods

The non-enterable objects that are placed in the Header, Break, and Footer areas are controlled by object methods. This section describes the methods used to print values in these locations.

Note The following code can be used only in object and form methods. It cannot be used in project methods.

The date is drawn from the system date by placing a non-enterable object named vDate in the Header area with this method:

vDate:= **Current Date**

The time is drawn from the system clock by placing a non-enterable object named vTime in the Header area with this method:

vTime:= **Current Time**

The subtotal for sales in the level 2 Break area is calculated and displayed in an object named vSalesProd with the following method:

vSalesProd:= **Subtotal** (Sales)

The subtotal for sales in the level 1 Break area is calculated and displayed in an object named vSalesCust with the following method:

vSalesCust:= **Subtotal** (Sales)

The total for sales in the level 0 Break area is calculated and displayed in an object named vSalesTotal with the following method:

vSalesTotal:= **Subtotal** (Sales)

Note that even though all three objects use the same calculation, they produce different results. Because they are placed in different Break areas, they are executed at different times and perform their calculations for different groups of records. For an explanation of Break levels, see the section "Reports with Breaks" on page 666.

vPage:= "Page " + **String** (**Current form page**)

The Current form page function returns the page number.

Printing Subforms, Pictures, and Text Fields

You can use subforms, picture fields, and text fields in a report. These objects can be set to print with either a fixed or variable frame. Fixed frame objects print within the confines of the object as it was created on the form. Variable frame objects expand during printing to include the entire contents of the object.

You cannot place more than one variable frame object side-by-side on a form. You can place non-variable frame objects on either side of a variable frame Picture or Text field provided that the variable frame object is at least one line longer than the object beside it and that all objects are aligned on the top. Otherwise, during printing, the contents of the other fields will be repeated for every horizontal slice of the variable frame object.

You cannot place objects on either side of a variable frame subform.

You can place variable objects in the Detail area only.

Printing Subform Areas

You usually use an output form to print records in a subform.

► To set the print option for a subform:

1 **In the Form editor, display the subform properties in the Property List.**

2 **Expand the "Print" theme and select a print option.**

Because there may be more records than can fit in the subform area, 4D provides the following three check boxes for controlling how records in a subform are printed:

- Variable
- Fixed (truncation)
- Fixed (multiple records)

If you select the **Variable** radio button, 4D expands or contracts the subform area to print all the records.

If you select the **Fixed (truncation)** radio button, 4D prints only the records that fit into the area of the subform. The form is printed only once and those records that are not printed are ignored.

If you select the **Fixed (multiple records)** radio button, the frame remains the same size, but 4D prints the form several times to include all the records.

Note You cannot place any objects on either side of a variable frame subform. Objects placed on either side of the subform will be repeated for every line of the subform.

For more information about subform options, refer to the "Subforms Options" paragraph on page 629.

Printing Picture Fields

Pictures can be printed with either fixed or variable frames if their display format allows it. A specific property is used to set the printing of pictures having one of the following display formats:

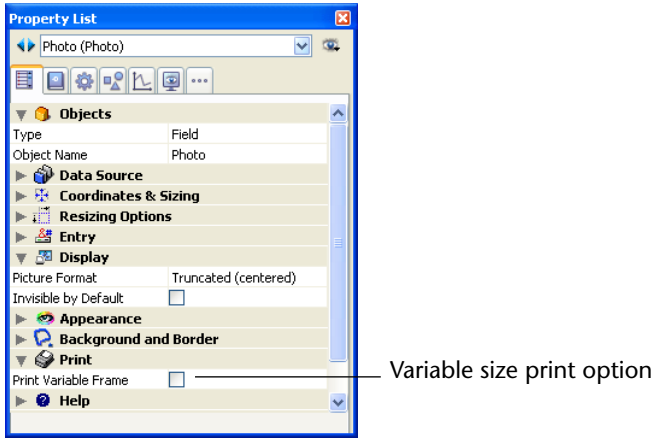
- Truncated (Centered)
- On Background
- Truncated (Non-centered).

For more information about these picture formats, refer to the "Picture Field Formats" paragraph on page 534.

► To set a print option for a picture field:

1 **In the Form editor, display the properties of the Picture field in the Property List.**

2 **In the "Print" theme, check (or uncheck) the Print Variable Frame option according to your specific needs:**



- If you check this option, the picture will be printed at a height that takes its size into account. The picture frame will be extended during printing if necessary in order to display the entire picture.
- If you do not check this option, the picture will be printed at a fixed height (set in the form).

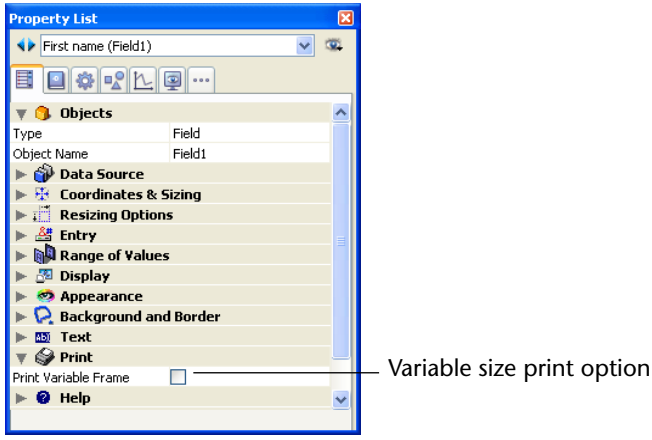
Printing Text Fields

Text fields can be printed with either fixed or variable heights. Print options for Text fields are selected exactly as they are for Picture fields.

► To select a print option for a Text field:

1 **In the Form editor, display the properties of the Text field in the Property List.**

2 **In the "Print" theme, check (or uncheck) the Print Variable Frame option according to your specific needs:**



- If you check this option, the text will be printed at a height that takes its size into account. The text field will be extended automatically during printing so that all the text it contains will be printed.
- If you do not check this option, the text will be printed at a fixed height (set in the form).

Printing Labels

You can generate labels with either the Label editor or a custom report form. If you use a report form, you have more extensive customization options. In particular, you can use the Form editor to insert variables within your labels.

Once you have created a label report form in the Form editor of 4D, you can use it in two ways:

- **Using the PRINT LABEL command**
  The PRINT LABEL(table) statement causes the printing of the current selection of *table* in the current List form. In this case, 4D uses the markers (width and height) of this form to set the label format.
- **Using the Label editor**
  This operation allows you to benefit from both the advanced functions of the Form editor and the layout parameters of the Label editor. The Label editor only takes the absolute position of the objects present in the form, as well as any margins set, into account. The positions of the markers are ignored.

Using the PRINT LABEL Command

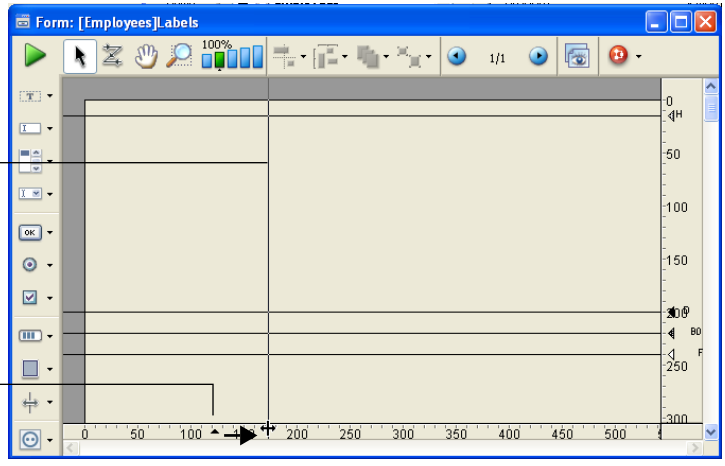
► To create a label report form and print it using the PRINT LABEL command:

1 Set the label width by dragging the width marker on the bottom ruler of the Form editor.

The width marker determines how many labels the PRINT LABEL command prints across the page. The placement of the width marker should correspond to the width of your labels. Be sure to take the left and right margins of your labels into account.

You can determine these margins by calculating the space between each label and dividing this number by two.

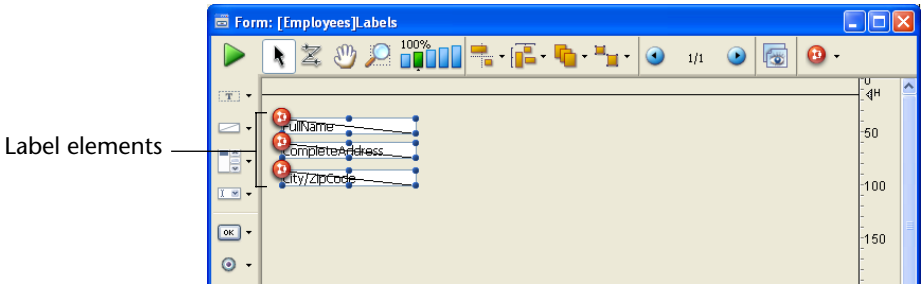
If you have labels that are 2.25 inches wide with a margin of .125 inches (1/8 inch) on either side of each label, the label width marker should be placed at 2.625 inches (2 3/8 inches) to ensure that the label text is placed correctly on the labels.



Note You may want to change the ruler units in the form to inches to make it easier to determine the proper placement of the label width marker. To change the ruler units, choose **Define Ruler Units** from the **Form** menu and select Inches.

2 Insert and set the objects making up each label.

Labels may contain fields and active objects (with which you can associate methods if desired), as well as graphic objects and static text. A subform cannot be printed in a label.



This label contains variables (active objects) whose values are calculated by object methods. For example, the method of the *FullName* variable concatenates the first and last name of each person and places a space between them:

```
FullName:=[Customers]FirstName+" "+[Customers]LastName
```

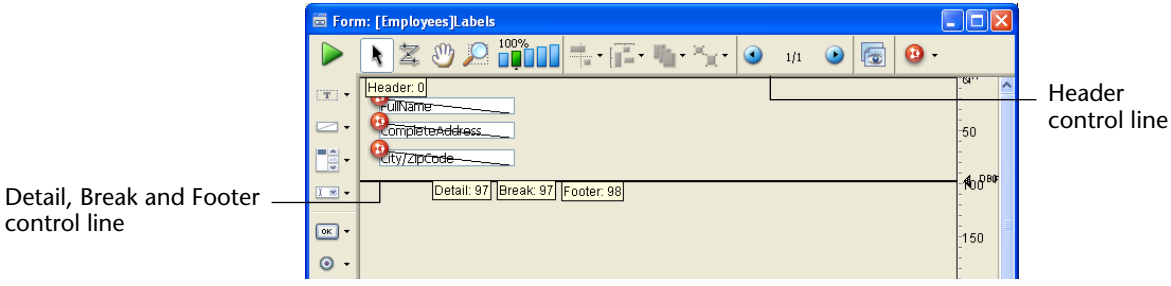
For more information about object methods, refer to chapter 11.

3 Position the objects in the form with respect to the width marker.

The label elements must be placed to the left of the width marker. They must be centered between the left edge of the form and the marker in order for the space on each side of the label to be the same.

4 Set the control lines so that the Header control line is above the label and the Detail, Break, and Footer control lines are below the label.

The Header control line should be set at zero inches and the Detail control line should be set to the height of the label. To center the text within the label, center the form elements between the Header control line and the Detail control line.



When you print the labels, everything between the Header and the Detail control lines appears on the labels.

5 **Save the form and print the selection using the PRINT LABEL command.**

For more information, refer to the *4D Language Reference* manual.

Using the Label Editor

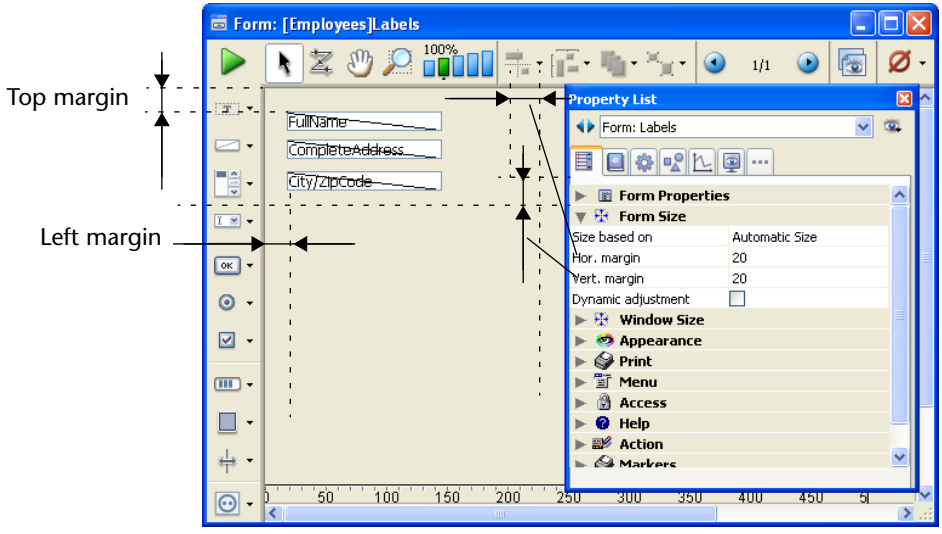
► To create a label report form and print it using the Label editor:

1 **In the form, insert and set the objects making up each label.**

Refer to step 2 in the previous paragraph.

2 **Position each element in the label according to the following principles:**

- The left and top margin of each label is determined by the position of the elements with respect to the (0,0) coordinates of the form.
- The right and bottom margin of each label is determined by the **Hor. margin** and **Vert. margin** values set in the Property List of the form.



3 **Choose the label form from the List of Tables window to make it the current output form for the table.**

If the List of Tables window is not already displayed, choose the **Tables List** command in the **Records** menu.

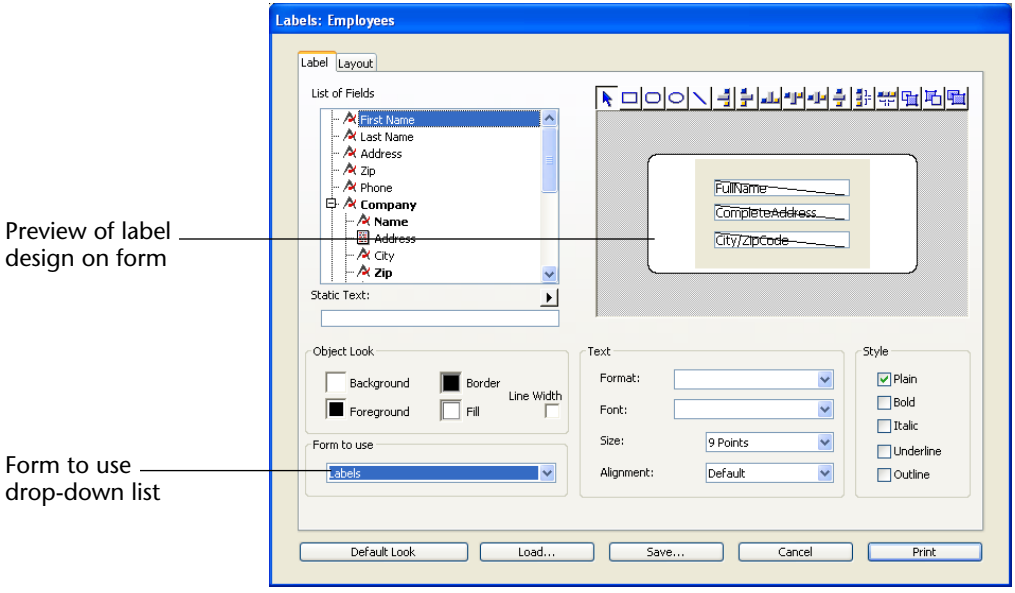
For more information about using the List of Tables window, refer to the "Browsing Different Tables and Forms" paragraph on page 947.

4 **Choose Labels from the Report menu.**

The Label editor appears.

5 **Choose the label form you designed from the "Form to use" drop-down list.**

This tells 4D to use this form to print the labels:



6 **Make any other changes to the Label editor such as setting the label margins or specifying the font in which the text appears.**

For complete information about using the Label editor, refer to the chapter "Labels" on page 1031.

7 **Click the Print button.**

4D prints the current selection of records using your label report design.

9 User Forms

In 4D, the developer can let users modify certain forms of the database in order, for example, to create custom reports, add a specific logo or, more generally, adapt the forms to their own specific needs. Once saved, a "user form" is substituted for the original form on the user machine and can be used like any other 4D form.

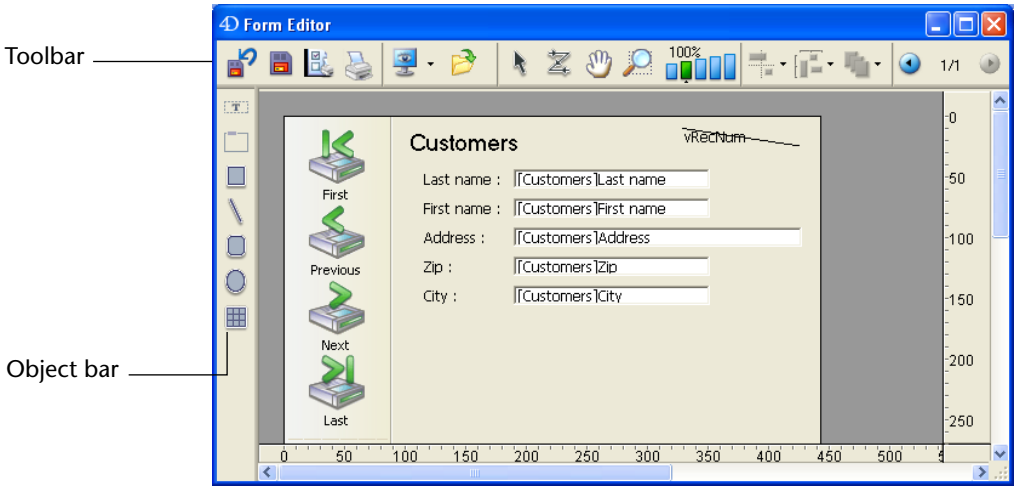
The possible modifications mainly concern graphic aspects. For example, the user cannot add object methods, variables or fields. They can, however, change the entry order or add active objects from an object library. The various possibilities are set up beforehand by the developer in the Form editor of the Design mode.

The mechanisms for setting up this system on the developer side (Design mode) are described in the "User forms" section of the *4D Language Reference* manual.
The current chapter describes the operation of the user form editor in Application mode.

User Form Editor

The User form editor is displayed for a form when the EDIT FORM command is executed, if the **Editable by user** property was checked for the form (see the "Editable by User" paragraph on page 402). Generally, this editor is called from a database configuration or customizing command.

The editor window appears as follows:



This editor uses the standard 4D Form editor interface, in particular the two integrated toolbar concept (tools and objects). It also has its own specific functions.

When the editor is displayed, only the **Edit** menu of the menu bar is available. It is still possible to display a simplified Property List as well as a contextual menu.

The selection and move object mechanisms (when allowed) are identical to those in the standard 4D Form editor.

For a description of the standard 4D Form editor, please refer to the chapter “Form Editor” on page 371.

Toolbar

The toolbar contains the following items:

Revert to factory settings: Returns to the original form as defined in the Design environment (deletes all changes made in the user form). When you click this button, an alert dialog box appears, allowing you to cancel or confirm the operation.

Save: Saves changes made in the user form.

Print settings and **Print**: These two commands allow setting and starting the printing of the user form.

Display: Allows selecting items to display and hide in the editor using a pop-up menu:

- **Page 0**: Shows or hides the objects from page 0 in the form's current page. This option allows you to distinguish between the objects on the form's current page and those on page 0. In a multi-page form, page 0 contains objects that are displayed on all the pages.
- **Paper**: Shows or hides the borders of the printing page, which are shown as gray lines. This option can have no apparent effect when the **Limits** (see below) option is selected. If the size of the form is smaller than the printing page, the page's borders are shown outside of the form's viewing area and therefore do not appear.
- **Rulers**: Shows or hides the rulers in the form editor's window.
- **Markers**: Shows or hides the output control lines and associated markers that show the limits of the form's different areas (header, body, footer). Markers are mainly used in output forms.
- **Limits**: Shows or hides the form's limits. When this option is selected, the form is displayed in the editor with its final dimensions.

Library: Allows loading the object library that can be used with the form (if any) (see the "Using an Object Library" paragraph on page 696).

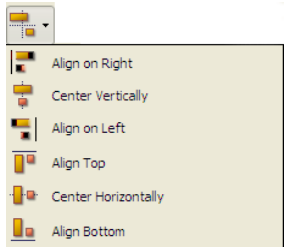
Selection tool: Allows selecting, moving and resizing form objects.

Entry order: Switches to "Entry order" mode, where it is possible to view and change the current entry order of the form (see the "Data Entry Order" paragraph on page 693).

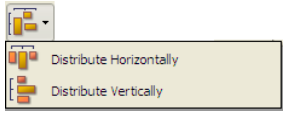
Moving: Passes into "Move" mode, where it is possible to quickly reach any part of the form by using drag and drop in the window. The cursor takes the shape of a hand. This navigation mode is particularly useful when zooming in the form.

Zoom: Allows modifying the form display percentage (100% by default). You can pass into "Zoom" mode by clicking on the magnifying glass or by clicking directly on the desired bar. When you click on this button, the cursor takes the shape of a magnifying glass . To increase the zoom percentage one level, click inside the form. To reduce the zoom percentage a level, press **Shift** (the cursor is changed to) and click inside the form.

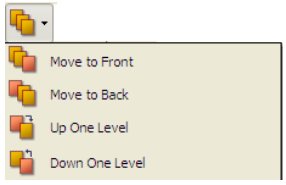
Alignment: This button is linked to a menu that allows aligning objects in the form. For this menu to be enabled, at least two objects must be selected in the form.



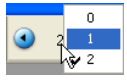
Distribution button: This button is linked to a menu that allows distributing objects in the form. For this menu to be enabled, at least three objects must be selected in the form.



Level button: This button is linked to a menu that allows changing the level of objects in the form.



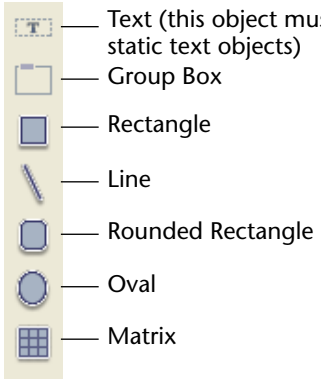
Display of pages: This area allows passing from one form page to another. To do so, click the arrow buttons or click the central area and choose the page to display from the menu that appears:



It is not possible to add pages in the user form editor.

Object Bar

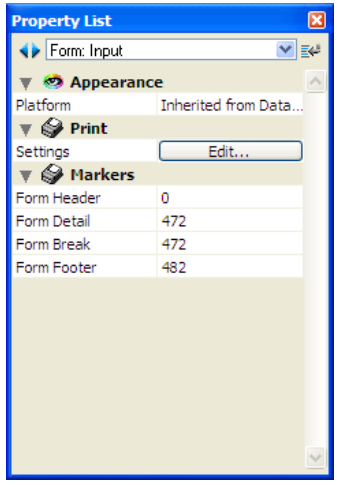
The object bar of the user form editor lets you add various graphic objects to the form:



To create an object in the form, select the corresponding button and then trace the object in the form.

Viewing and Editing Objects

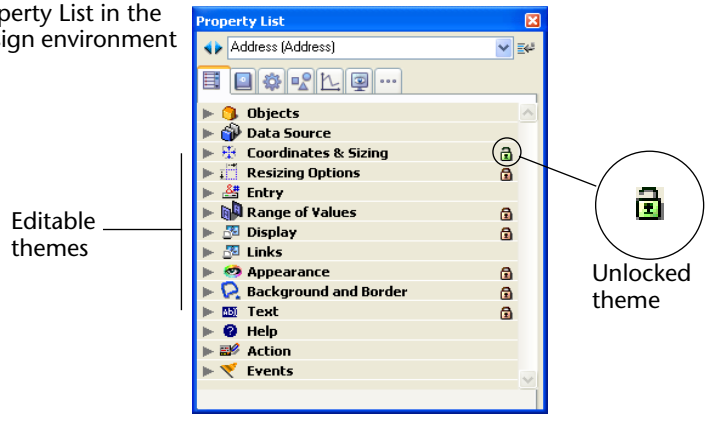
The user can view and edit the properties of objects present in the form using a simplified Property List. To display the list, simply **double-click** the object or choose the **Property List** command in the contextual menu of the editor (right-click).



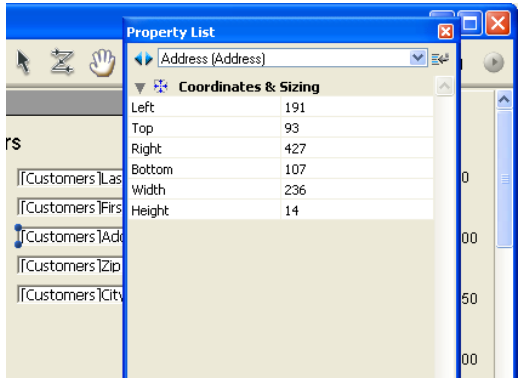
Only user-editable properties are displayed in the Property List.

- For graphic type objects added by the user using the object bar, all properties linked to the object type are available.
- For objects coming from the original form, two conditions must be met in order for a property to be editable in the User form editor:
  - The property must belong to a set of properties that can be declared editable for the object or form. For example, the properties related to managing form events or drag and drop are never editable by the user.
    In the Form editor of the Design environment, the editable property themes are distinguished with a padlock icon in the Property List (see the "Editable by User" paragraph on page 402).
  - The property theme must have been explicitly declared as editable by the developer for the object or form. To do that, the developer must have clicked on the padlock in the Property List of the Form editor to unlock it:

Property List in the Design environment



The unlocked properties then become accessible by the user in the User form editor:



This also works for other possible actions in the user form: for example, if the “Coordinates & Sizing” theme is unlocked, the user could move the object using the mouse or the keyboard.

Edit Menu Commands

Only the **Edit** menu is active when the User form editor window is displayed in the Design environment. When using customized interfaces in the Application environment, the **Edit** menu is accessible if the commands it contains are managed using standard actions.

All editing commands (**Cut**, **Copy**, **Paste**, **Clear**, **Select all**) work without restrictions for objects added by the user.

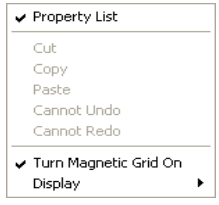
On the other hand, for reasons of consistency, the user is never allowed to remove an object from the original form. The use of the **Cut**, **Clear** commands or the **Del** or **Backspace** keys cause the display of an alert dialog box telling the user that this action is not allowed.

Contextual Menu

Several commands are accessible using the User form editor contextual menu. It is not possible to control the contents of this menu.

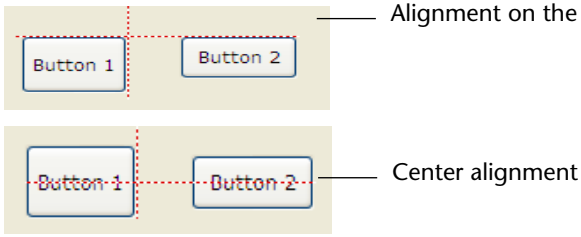
Clicking Outside of Any Objects

When the user clicks outside of any object, the standard display and editing commands of the form editor are available: display of the Property List, Magnetic grid activation and display of the editor tools:



- **Turn Magnetic Grid On**: The magnetic grid offers you a visual aid for aligning and distributing objects in the form.
  The magnetic grid works as follows: when you drag and drop an object in the form, 4D indicates the possible locations for this object based on alignments relative to other form objects:
  - Horizontally possible locations are suggested when the edges or centers of two objects coincide,
  - Vertically possible locations are suggested when the edges of two objects coincide.

At this moment, 4D places the object at the location and displays a red line indicating the noticeable alignment taken into account:



Concerning the distribution of objects, 4D proposes a distance based on interface standards. Like with magnetic alignment, red lines indicate the noticeable differences once they are reached.

Note You can manually activate or deactivate the magnetic grid when an object is selected by pressing the **Ctrl** (Windows) or **Control** (Mac OS) key.

- **Display**: This menu has the same items as the Display button in the editor toolbar (see the "Toolbar" paragraph on page 686).

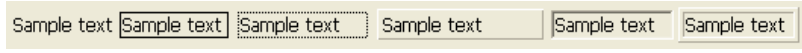
Clicking on an Object

When the user clicks on an object, graphical commands are available (color, alignment, etc.).

Access to these commands can be restricted for objects that belong to the original form, depending on the type of object clicked as well as the settings made by the developer in the Form editor.

- **Color**: Displays two palettes for modifying the foreground and background colors of the object.

- **Align**: Provides access to commands for aligning and distributing objects. The alignment commands are only enabled if at least two "movable" objects are selected. The distribution commands are only enabled if at least three "movable" objects are selected.
- **Border Line Style**: Used to modify the border line style of the selected objects. Here are examples of the styles available:



Transparent Plain Dotted Raised Sunken Double

- **Automatic Size**: This option is used to resize text objects and pictures in order to obtain optimal display in keeping with their current graphic characteristics (font size, style, etc.) or to display them in their original size (pictures).
  You can obtain the same result by pressing **Ctrl** (Windows) or **Command** (Mac OS) + clicking on the lower right-hand corner of the object.
- **Level**: Used to change the level of objects.
- **Select Similar Objects**: Used to select all the objects that are the same type as the selected object.

Data Entry Order

The data entry order is the order in which fields, subforms, and other tabable objects are selected as you tab through a form.

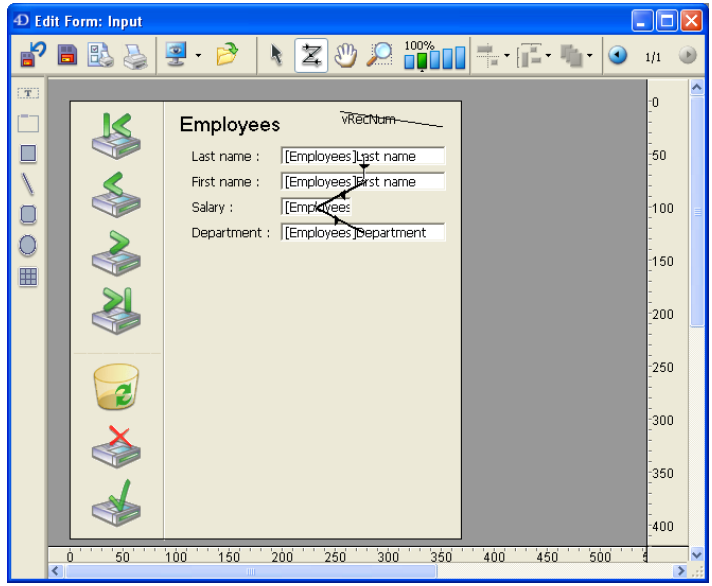
Note For more information about default data entry order in forms, please refer to the "Data Entry Order" paragraph on page 475.

In some cases, it may be necessary to modify the data entry order of a form, for example when you have added one or more objects and want to include them in a more logical order.

Viewing and Changing the Data Entry Order

To view the current data entry order of a form, click the **Entry order** button in the toolbar of the user form editor:

The current data entry order appears as follows:



- To change the data entry order, position the pointer on an object in the form and, while holding down the mouse button, drag the pointer to the object you want to follow in the data entry order. 4D will adjust the data entry order accordingly.
- When you are satisfied with the data entry order, click the **Entry order** button in the toolbar.

Setting the First Object in the Data Entry Order

All enterable objects are part of the data entry order. To define the first object of the entry order, the Entry Order mode must be disabled.

► To establish one of the objects as the first in the data entry order:

1 **Select the object you want to be first in the entry order and move it to the back.**

You can use the **Level** > **Move to Back** command in the contextual menu or the corresponding command of the button in the editor toolbar.

The object will be placed in back of any other form element. This is temporary.

2 **Click the Entry order button of the toolbar.**

The selected object becomes the first object in the entry order and the object that was first becomes second. You can now drag from object to object in order to create the data entry order you want.

3 **When you have finished setting the entry order, click on the Entry order button of the toolbar again.**

4D returns to normal operation of the form editor.

The object you selected to make first in the entry order has returned to its previous location (it is no longer moved to the back of the form).

Using a Data Entry Group

While you are changing the data entry order, you can select a group of objects in a form so that the standard data entry order applies to the objects within the group. This allows you to easily set the data entry order on forms in which fields are separated into groups or columns.

► To create a data entry group:

1 **Click the Entry order button of the toolbar.**

2 **Draw a marquee around the objects you want to group for data entry.**

When you release the mouse button, the objects enclosed or touched by the rectangle follow the standard data entry order. The data entry order for the remaining objects adjusts as necessary.

Restoring the Standard Data Entry Order

You can restore the standard data entry order (based on the layering of the objects) at any time.

1 **Click the Entry order button of the toolbar.**

2 **Draw a marquee around all the objects on the form.**

When you release the mouse button, 4D re-establishes the standard data entry order for all the objects enclosed or touched by the rectangle.

Using an Object Library

It is possible to use an object library with each user form being modified.

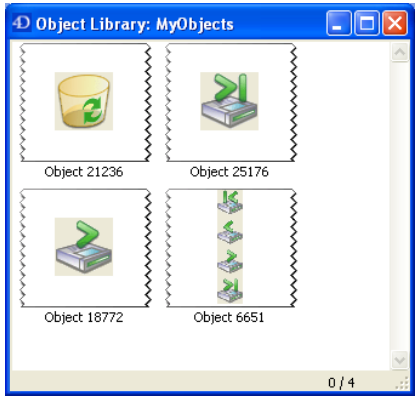
An object library is an external file created by 4D in the Design environment. It allows storing all types of objects that can be used in forms: buttons, texts, pictures, hierarchical lists, etc. Libraries are put together and used by a simple drag and drop or copy-paste from 4D forms. They are somewhat like a permanent clipboard.

Note For more information about object libraries, please refer to the "Using Object Libraries" paragraph on page 461.

Objects are stored with all their properties and standard actions. It is not possible to use objects with associated object methods in the user form editor. Only objects that do not have associated object methods will be visible in the library from the user form editor.

The developer sets the library associated with the form using the last parameter of the EDIT FORM command which displays the user form. If this parameter is valid, the library can be displayed and used with the user form. To open the library, the user must click on the corresponding icon in the toolbar of the editor:

The library window is then displayed:



Note An object library can contain up to 32,000 items.

The user can add objects to the form by drag and drop or copy-paste from the library.
All the properties of objects inserted from the library are automatically accessible to users (coordinates, dimensions, appearance, etc.).

Each object library has a contextual menu that allows accessing basic editing commands. In the user form editor, only the **Copy** command is available. Moreover, it is not possible for the user to remove, rename or add objects in the library.

10 Quick Reports

Introduction

One of the most important tasks in data management is report generation. The Quick Report editor is one of two tools made available by 4D for the purpose of designing reports. You use the Quick Report editor to create ad-hoc reports in the Design and Application environments. The other tool is the Form editor, which you can use to design reports in the Design environment (which can be altered subsequently in Application mode). You should use an output form to design reports that require complex designs or programmatic processing. This possibility is detailed in the chapter "Output Displays and Reports" on page 635.

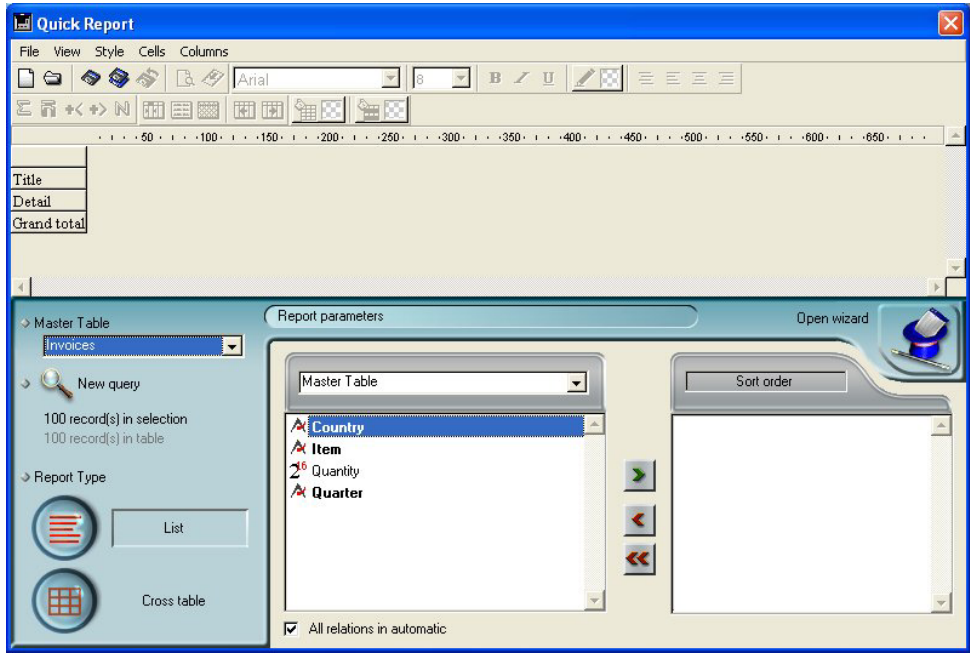
Using the Quick Report editor, you can:

- Produce lists of records.
- Create break areas.
- Produce cross-table reports.
- Compute summary calculations.
- Modify fonts and styles in the report.
- Define borders and background colors on a cell basis.
- Save and open quick report designs to disk.
- Select different output types such as HTML or text file, 4D View or 4D Chart area, printing or saving to disk.

  The Quick Report editor produces reports from the current selection of records. Before generating a report, set the current selection of records that you want to include in your report.

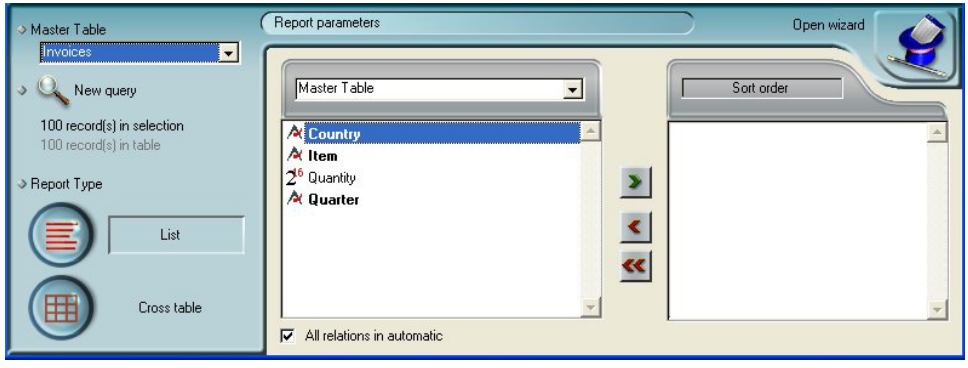
Overview of the Editor

The Quick Report editor appears as follows:

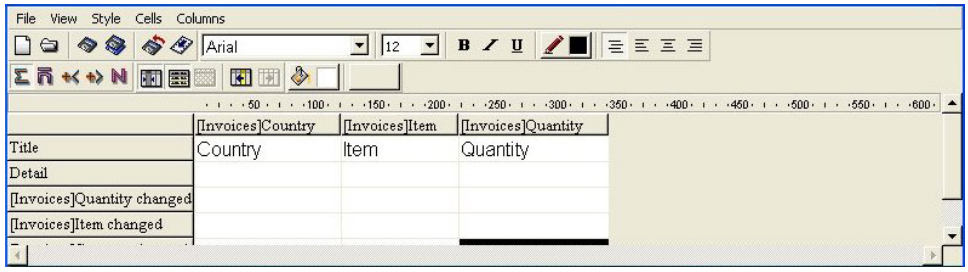


It consists of two main areas:

- The Quick Report data area that is used to specify the type of report to be built and the data to be integrated; it also can be used to access the Quick Report Wizard:

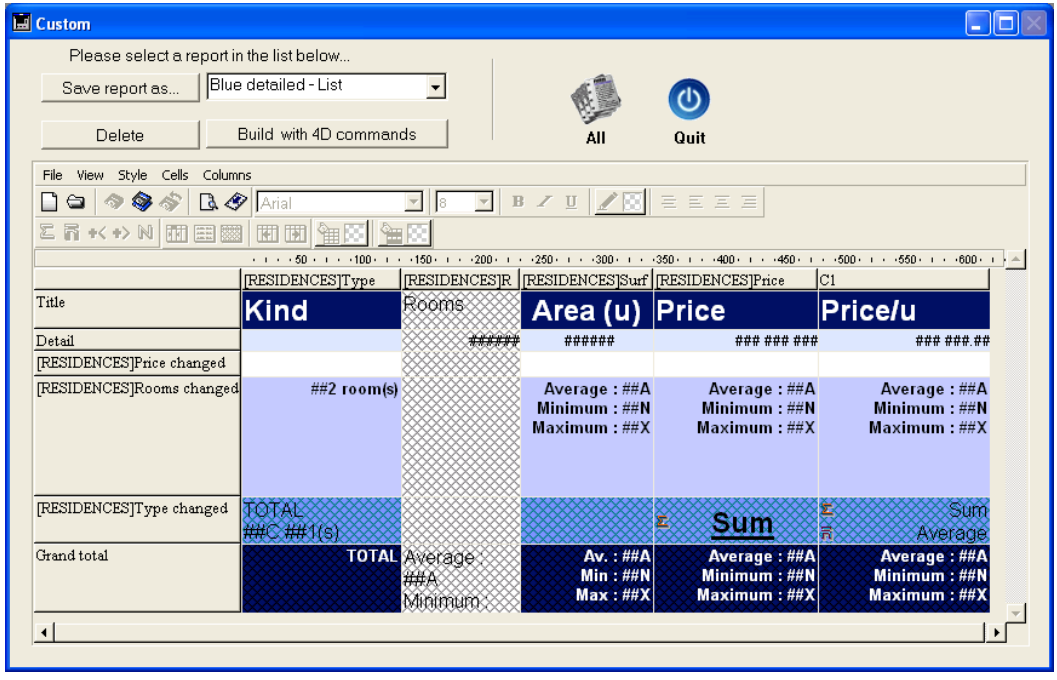


- The actual Quick Report area:



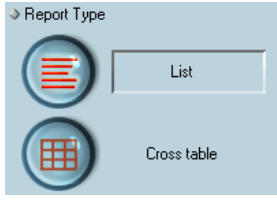
This shows the area as it appears when a Quick Report area is inserted in a form. This is where actual column and row contents are created, where sort orders are defined, etc. Also, this area includes various toolbars, its own menu bar and contextual menus.

For example, this area could be placed in a form as depicted below:



List Mode and Cross-table Mode

In the Quick Report editor there are two operation modes available which generate two specific types of reports: **List** and **Cross table**.



- **List mode**
  This is the default mode. In this mode, reports would typically display records as a list with break levels where sums would be performed. The following is a typical List quick report.

First Name	Last Name	Department Name	Salary
Biff	Davis	Accounting	43780
Smeldorf	Garbando		19610
Alan	Hull		41460
Bryan	Pfaff		26440
Shirley	Ransome		36040
Marlys	Wilson		36500
		Sum for Department : Accounting	203830
Kathy	Forbes	Engineering	18840
Dennis	Hanson		40520
Mary	Smith		55000
Andy	Venable		43520
Lance	Wolfram		27300
		Sum for Department : Engineering	185180

- **Cross-table mode**
  This mode allows you to display your report as a two-dimensional table. This is useful when you want to display data from a data source broken down into categories that are actually a function of two other data sources.

For example, a cross-table form would let you display in a table how many of each product type was sold in each quarter. The following is a typical Cross-table Quick Report.

	AV Preamplifier	Power module	Remote control	Line Total
Q1	34	29	39	102 10526316
Q4	48	64	21	133 66666667
Q3	49	68	40	157 48148148
Q2	64	74	47	185 66666667
Grand total	195 5.909090909091	235 5.875 1	147 5.4444444444 1	577 5.77 1

Quick Report Basics

This section describes basic operations used when working with the Quick Report editor to design a report.

Creating a New Quick Report

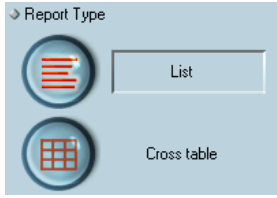
The Quick Report editor lets you create reports in the Design or Application environments.
It is also possible to insert Quick Report area into a form via a Report type plug-in area (see the "Plug-in Areas" paragraph on page 592).

► To design a quick report:

1 Choose Quick Report... from the Tools menu or click on the Tools>Quick Report button in the toolbar.

4D displays the Quick Report editor. If a previous report has been defined, it appears in the window. To erase the contents of the window, choose **New** from the **File** menu or click the New icon in the standard toolbar of the editor  to begin a new Quick Report design.

2 Choose the type of report to be generated by clicking on either the List or Cross table button, located at the bottom left corner of the Quick Report editor:



The List type is generally selected by default. The Quick Report area is modified according to the type selected.

Loading and Saving a Quick Report Design

You can save a quick report design as a file that you can open from the Quick Report editor. The quick report design includes all of your specifications for the report, but not the data. By saving report designs, you can maintain a library of quick report designs that you can use according to your needs.

Note If you do not save your design, it is displayed the next time you open the Quick Report editor but only during the same work-session.

Saving a Quick Report Design

► To save a quick report design:

1 Choose Save as from the File menu.
OR
Click the Save as icon in the file management toolbar  **.**

4D displays a create-file dialog box in which you can name the quick report design.

Note Quick report files in 4D for Windows are denoted by the file extension .4QR.

2 Enter a filename for the quick report and click OK.

4D saves the report as a file that you can open with the Quick Report editor. You can save the file on your hard disk. If you modify the parameters of the report subsequently and want to save them again, you just need to choose the **Save** command in the **File** menu.

The new file will replace the previous one on the disk, without the standard save file dialog box appearing.

Loading a Quick Report Design

When the Quick Report editor is open, you can load a saved design and use it to print a new report. You can use the same quick report design repeatedly to print different selections of records.

► To load a report design:

1 **Choose Open from the File menu.**
OR
Click the Open icon in the file management toolbar  **.**
4D displays an open-file dialog box displaying a list of available quick report designs.

2 **Double-click a filename or select a filename from the list and click OK.**

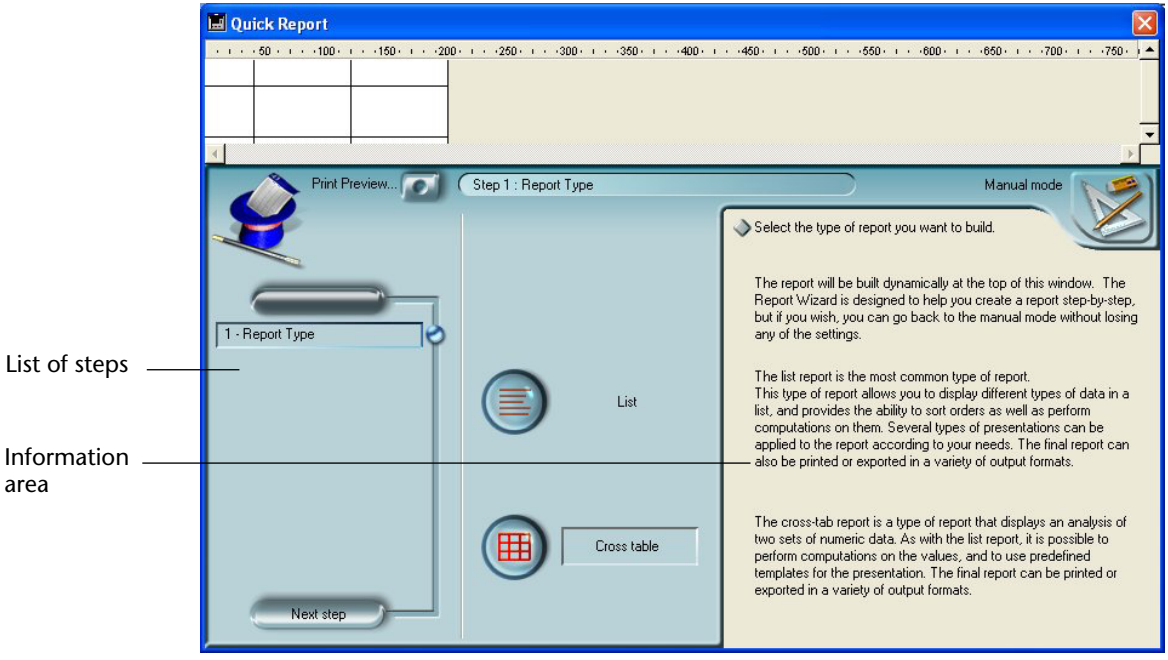
4D replaces the current design with the design you opened.

The Quick Report Wizard

When creating a manual Quick report, you can use the Wizard to guide you through the creation of the report.

To start the wizard, click the **Open wizard** button in the Quick Report editor:

The pages of the wizard then replace the data area in the window of the Quick Report editor:



The construction of the report is then carried out through detailed steps with comments appearing in the right-hand part of the window. To create a Quick Report using the wizard, follow the instructions displayed on the screen.

When you have finished constructing the report, the wizard offers you the possibility of generating the corresponding 4D code. For more information about this, refer to the “Generating 4D Code” paragraph on page 754.

To exit the wizard, click on the button to return to Manual mode:

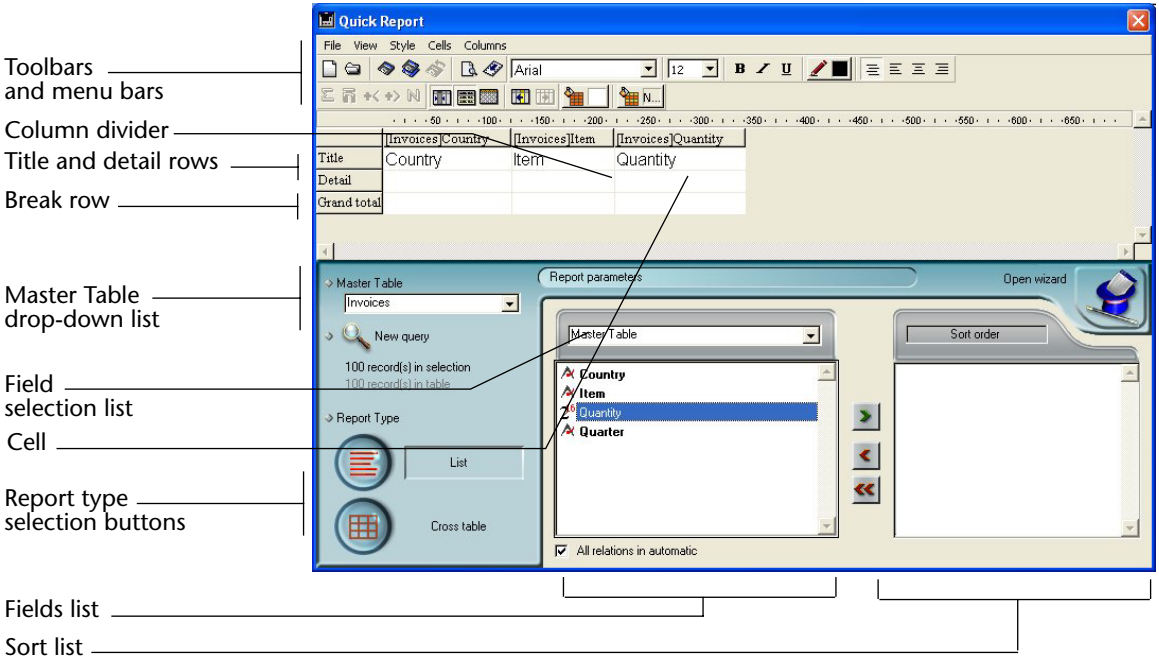
Using the Quick Report Editor

Interface

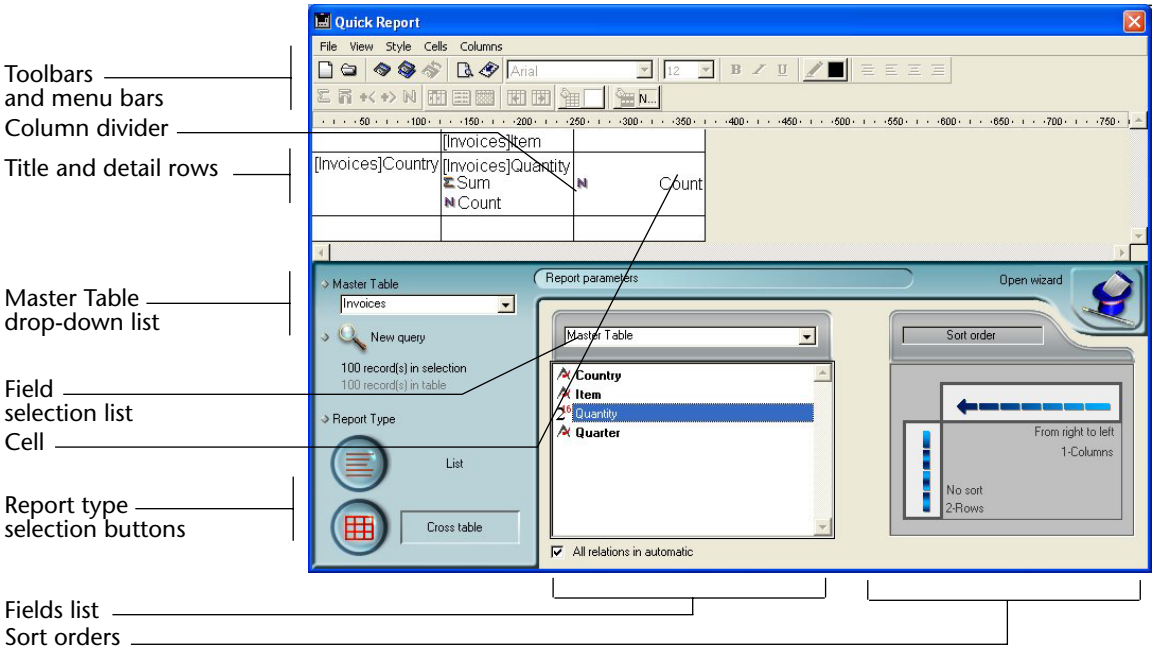
When you create a quick report, you can specify the following:

- Columns that display fields or formulas, either from the current table or from related tables,
- Sort levels and order (list mode),
- Summary calculations,
- Display format,
- Text for labels,
- Formats for numeric and Boolean data,
- Font, font size, style, and justification for labels, summary calculations, and data,
- Background colors on a cell column or row basis,
- Borders,
- Page headers and footers.

List mode:



Cross-table mode:



The Quick Report editor contains the following elements:

- **Master Table drop-down list**: This menu is used to specify the master table that will be used as a basis for generating the report. The fields of this table are displayed in the Fields list and the related fields will be displayed in relation to this table.
- **Fields list**: This list lets you select the fields to be inserted into the report by double-clicking or by drag and drop. It displays either the list of fields found in the master table, the hierarchical list of tables and related fields, or the hierarchical list of all the tables and fields of the database. You choose the type of display in the Field selection list located just above the area.
  Indexed fields appear in bold. You can also display and select the fields of related tables.

Note Only tables and fields which are not hidden appear in the Fields list of the Quick Report editor. For information about hiding tables and fields, refer to the "Invisible" paragraph on page 261.

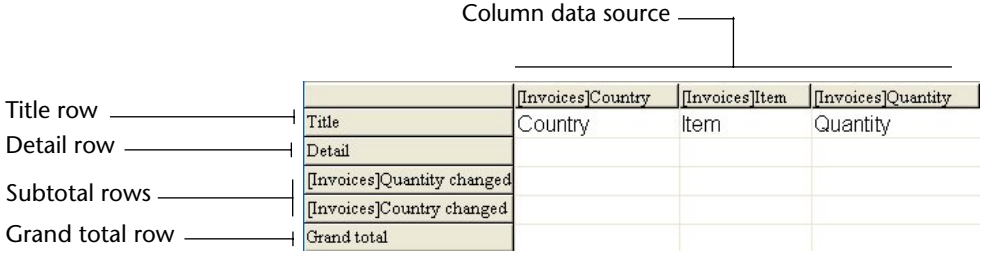
- **All relations in automatic**: This option is used to specify the way in which the Quick Report editor will make use of the relations between the different tables of the database.
  On principle, the editor can only use automatic relations. By default, the option is not checked and only automatic relations can be used by the editor.
  If the **All relations in automatic** option is checked, the Quick Report editor considers the manual relations of the database as automatic relations and will thus enable access to all the data related to the database.

Note to developers

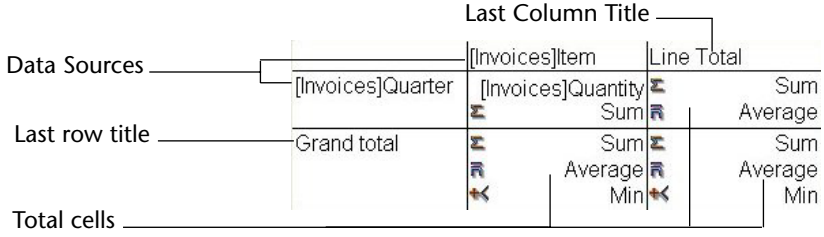
• The **All relations in automatic** option causes all the relations to pass into automatic for the current process, even after the Quick Report editor has been closed. If you want to make sure that all the relations will be manual after using the Quick Report editor, execute the statement AUTOMATIC RELATIONS(False;False).
• When the editor is called using the QR REPORT command, this option does not appear in order to allow the developer to manage the status of relations directly.

- **Column dividers**: These lines indicate the boundaries between columns of the report. They can be moved manually in order to enlarge or reduce the size of each column. Manual resizing deselects the **Automatic Width** option if it has been activated for the column concerned.
- **Cells**: A cell is the intersection of a row and a column.
- **Scroll bars**: You use the scroll bars to view parts of the quick report design that extend beyond the area of the quick report form.
- **Sort list/Sort order display area**:
  - In List mode, this list displays the fields of the report on which the sort will be carried out, as well as the sort order and whether it will be ascending or descending. Each field inserted into this list causes a sub-total row to be added in the Quick Report area.
  - In Cross-table mode, this area displays the sort order for each data source.

- **Quick Report area**: This area lets you build your report by inserting fields using drag and drop, double-clicking or via the contextual menu; you can also adjust the width of the columns added, or delete breaks or formulas, define the colors and borders of cells, etc.



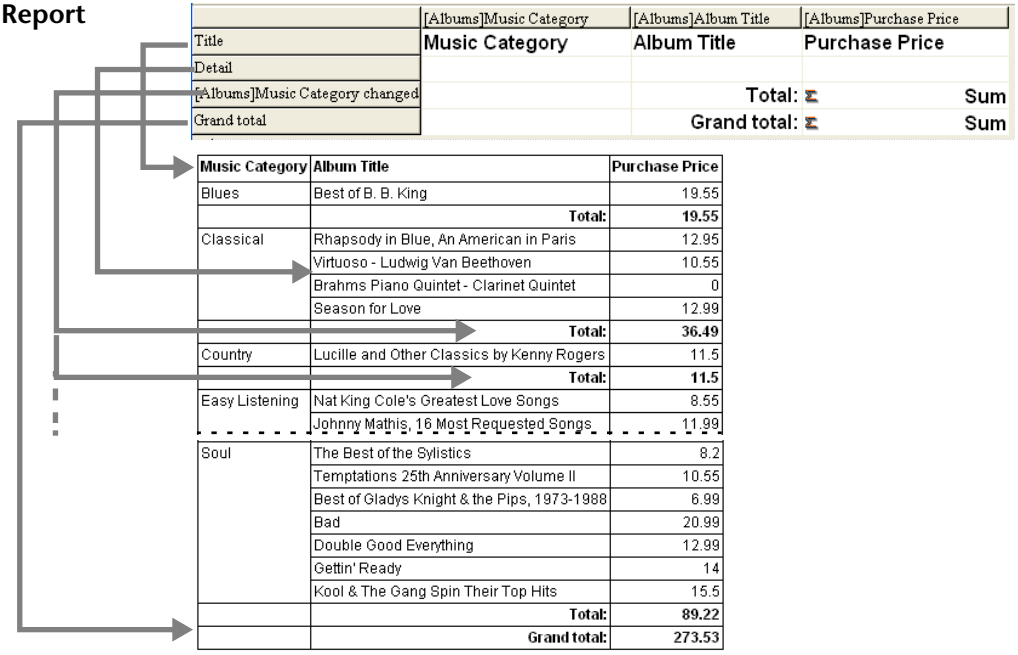
- **Title row**: This row displays the names of fields or formulas that have been inserted into the report. It is repeated for each page of the report. The Quick Report editor inserts field names by default, but you can modify the contents.
- **Detail row**: This row contains information drawn from each record and is repeated in the report for each record. You can associate a display format with it, depending on the type of data represented.
- **Subtotal rows**: These rows display intermediate calculations as well as the wording that is associated with them. A row is created for each sort order.
- **Column data sources**: These titles indicate the source of the data for each column.



- **Data Sources**: These two cells will house fields that will be used for the two categories of the array. The center cell can also accept fields sources and calculations at the same time.
- **Total cells**: These cells are designed to house calculations on the column's contents.
- **Title cells**: These cells house the titles for the last column or the last row. Their contents can be modified.

The following figures shows completed quick report designs and the relationship between the specifications in the design and the printed output:

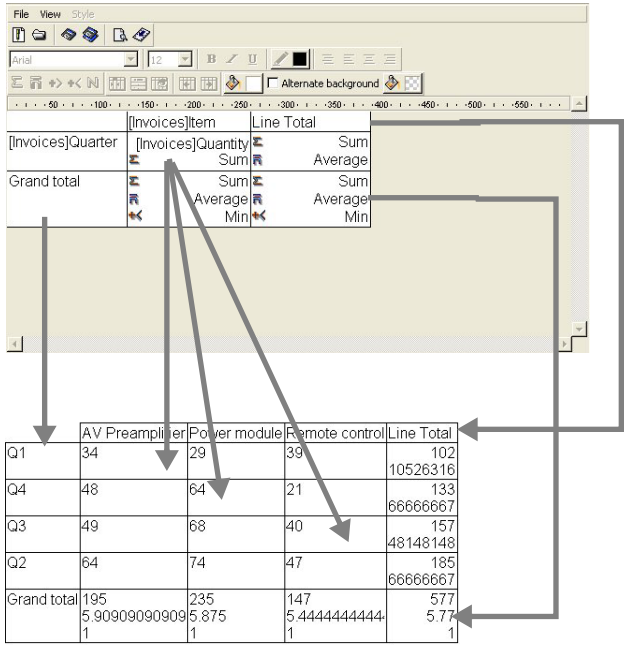
List Report



	[Albums]Music Category	[Albums]Album Title	[Albums]Purchase Price
Title	Music Category	Album Title	Purchase Price
Detail			
[Albums]Music Category changed		Total: Σ	Sum
Grand total		Grand total: Σ	Sum

Music Category	Album Title	Purchase Price
Blues	Best of B. B. King	19.55
	Total:	**19.55**
Classical	Rhapsody in Blue, An American in Paris	12.95
	Virtuoso - Ludwig Van Beethoven	10.55
	Brahms Piano Quintet - Clarinet Quintet	0
	Season for Love	12.99
	Total:	**36.49**
Country	Lucille and Other Classics by Kenny Rogers	11.5
	Total:	**11.5**
Easy Listening	Nat King Cole's Greatest Love Songs	8.55
	Johnny Mathis, 16 Most Requested Songs	11.99
Soul	The Best of the Stylistics	8.2
	Temptations 25th Anniversary Volume II	10.55
	Best of Gladys Knight & the Pips, 1973-1988	6.99
	Bad	20.99
	Double Good Everything	12.99
	Gettin' Ready	14
	Kool & The Gang Spin Their Top Hits	15.5
	Total:	**89.22**
	Grand total:	**273.53**

Cross-table Report



	[Invoices]Item	Line Total
[Invoices]Quarter	[Invoices]Quantity	Sum / Average
Grand total	Sum / Average / Min	Sum / Average / Min

	AV Preamplifier	Power module	Remote control	Line Total
Q1	34	29	39	102 10526316
Q4	48	64	21	133 66666667
Q3	49	68	40	157 48148148
Q2	64	74	47	185 66666667
Grand total	195 5.90909090909 1	235 5.875 1	147 5.4444444444 1	577 5.77 1

Contextual Menus

The Quick Report editor has contextual menus that make it easy to access certain row, column, and cell operations. Instead of making menu selections or working with the Cell or Column properties areas, you can perform certain operations by displaying a Quick Report contextual menu.

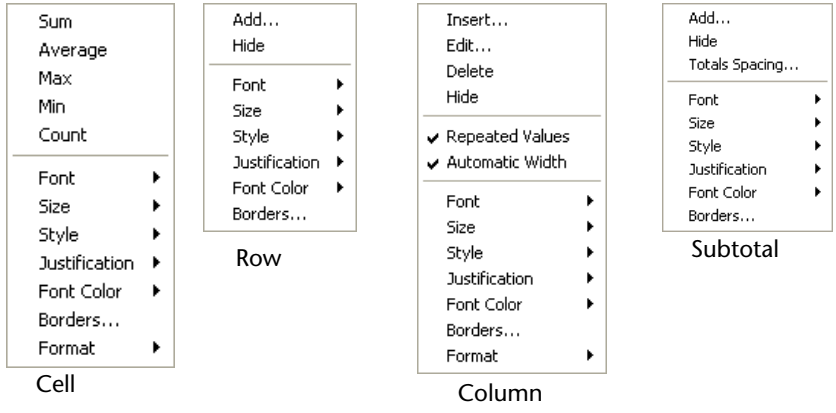
There are separate contextual menus for row, column, and cell operations.

► To use a contextual menu:

1 **Click with the right mouse button on a cell, a row title, or a column heading.**

A contextual menu appears. The commands in the contextual menu depend on where you click (column heading, cell, row or subtotal row title).

2 **Choose the desired menu command:**



As with any contextual menu, this menu's contents vary depending on which cell type is highlighted. Items available break down into five main types:

- Font attributes,
- Display formats,
- Standard calculations,
- Column or row management items, and
- Cell borders.

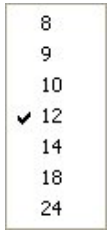
Font Attributes

These items appear in the contextual menu when a cell, column or row is selected. Selecting a font attribute will apply it to the current selection (cell, row or column). The Font attribute items appear as follows:

- **Font**
  Selecting this item displays the list of fonts installed on your machine from which you can select the font that will be applied to the items selected in the report area.

  Once an item in the list is selected, a check mark is displayed next to it.

- **Size**
  Selecting this item displays the list of font sizes from which you can select the font size that will be applied to the items selected in the report area.



  Once an item in the list is selected, a check mark is displayed next to it.

- **Style**
  Selecting this item displays the font styles (Plain, Bold, Italic, and Underline) from which you can select the font style that will be applied to the items selected in the report area.

  Once an item in the list is selected, a check mark is displayed next to it.

- **Justification**
  Selecting this item displays the list of justification attributes (Default, Left justified, Centered and Right justified).

Note Default is the default justification for this data type (left for numbers or, otherwise, right).

Once an item in the list is selected, a check mark is displayed next to it.

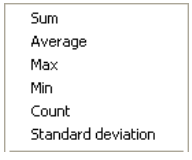
- **Font color**
  Selecting this item displays a palette of colors.

Display Formats

These items appear in the contextual menu when either a cell or a column is highlighted. From the menu, you can select one of the 4D default formats that match the data type of the column. Once selected, the format applies to the entire column, regardless of whether the current selection is a cell or a column.

Summary Calculations

These commands appear in the contextual menu when the current highlighted cell belongs to either the Grand total row or to a Subtotal row.



Selecting an item will insert the corresponding calculation in the highlighted cell (see the "Adding Summary Calculations" paragraph on page 737). Once an item in the list is selected for a cell, a check mark is placed next to it. It is possible to select several calculations successively.

Column and Row Management Items

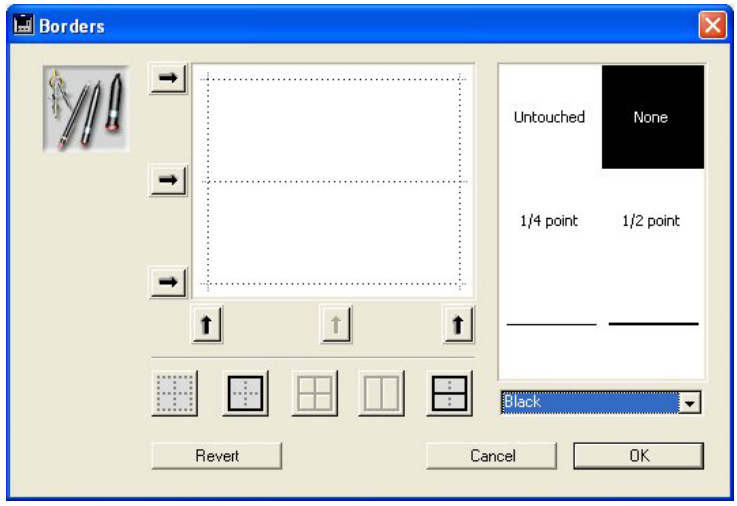
There are several items that are dedicated to column or row management:

- **Hide**
  Selecting this item hides the current column or row. Once it is hidden, a check mark is displayed next to the item, and the row or column is crossed out.
- **Add...**
  This item is displayed when a row is highlighted or nothing is highlighted. Selecting this item displays the formula editor to allow you to define the data source for a new column. Once this is done, the new column is added to the right of the right-most existing column.
- **Totals Spacing...**
  This item is displayed when you click on the title area of a subtotal row. It displays the Subtotals Properties dialog box. The options found in this dialog box are described in the "Subtotal Spacing" paragraph on page 736.
- **Insert...**
  This item is displayed only when a column is highlighted in list mode. Selecting this item displays the formula editor to allow you to define the data source for a new column. Once this is done, the new column is added to the left of the current highlighted column.
- **Delete**
  This item is displayed only when a column is highlighted in list mode. Selecting this item deletes the current highlighted column.
- **Edit...**
  This item is displayed only when a column is highlighted in list mode or when a cell is highlighted in cross-table mode. Selecting this item displays the formula editor to allow you to edit the data source for the current column (list mode) or for the current cell (cross-table mode).

- **Automatic Width**
  This item is displayed only when a column is highlighted in list mode, or when any cell is highlighted in cross-table mode. You can use this item to select or deselect the automatic width. When Automatic Width is selected, 4D will try to match the column size to the data so that the data fits on one line.

Borders

These items appear in the contextual menu when a cell, column or row is selected. Selecting Borders... displays the Borders dialog box:



For more information about this dialog box, refer to the "Setting Borders" paragraph on page 727.

Once defined, the border format is applied to the current selection (cell, row or column).

Selecting Rows, Columns, and Cells

When designing a List quick report, you need to select rows, columns, and cells in the quick report form. A cell is the intersection of a row and a column.

► To select a row:

- **Click on the Title, Detail, Subtotal, or Grand total cells in the row label area.**

▶ To select a column:

■ **Click the Header row of a column.**

▶ To select a cell:

■ **Click on the cell.**

Adding and Modifying Text

You can add or modify text in the quick report form to label parts of the report. For example, if you requested summary calculations, you can label them by adding text to other cells in the Subtotal and Grand total rows.

You can add and modify text as follows:

- Edit the text that 4D automatically adds to the Title row of the report,
- Insert text in empty cells of the Subtotal and Totals rows,
- Insert the value of a Subtotal field in the Subtotal rows,
- Specify the font, font size, justification, and style for any text that appears in the report.

Adding Text

▶ To add text in a report cell:

1 **Click twice on an empty cell in the quick report form.**
A text insertion point appears in the cell.

If you are entering a label for a summary calculation, select a cell in the same row as the cell containing the calculation icons. You cannot enter text into the same cell that contains summary calculations.

2 **Type the text in the cell.**

Modifying Text

▶ To modify text in a cell:

1 **Double-click in a cell to get an insertion point and drag across the text in the cell you want to modify.**
4D highlights the selected text.

2 **Type the new text in the cell.**

Adding Columns

In List mode, you create columns by dragging field names from the Fields list into the quick report area. If you want to add a field from a related table, expand the foreign key field in the hierarchical list of fields to display the fields in the related tables (if you have selected **Related Tables** or **All Tables** in the drop-down menu located just above the list).
You can add fields from related tables, provided that the relationship is automatic. If you want to be able to use manual relations, check the **All relations in automatic** option: in this case, the editor changes all the relations of the database to automatic (see the "Interface" paragraph on page 707).

Note You cannot add or edit a column in cross-table mode, since the report comes with all the columns needed.

► To add a field:

■ **Drag the name of a field to the right of existing columns in the Quick Report area and release the mouse button.**

OR
Double-click on the name of a field in the Fields list.

Note If you drag and drop a field onto an existing column, it will be replaced by the new column.

4D creates a column for the field and places the field name in both the column header and the cell in the Header row.

By default, 4D prints the field names as column heads at the top of each page in the quick report.

► To add a column using a menu command:

1 **Select the Add... command in the Columns menu of the editor.**
OR
Right-click over an unused area of the Quick Report area, then select the Add... command.

The standard 4D formula editor is displayed, allowing you to specify the data source of the new column (field or formula).

Inserting Columns

In List mode, you can insert an empty column into a quick report. After you insert the column, you can assign a field or a formula to it.

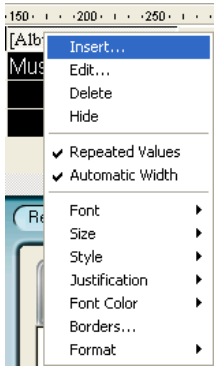
► To insert a column:

1 **Select a column.**

2 **Choose Insert... from the Columns menu.**

OR

Right-click to display the Quick Report contextual menu and choose Insert.



The standard 4D formula editor is displayed, allowing you to specify the data source of the new column (field or formula).

Deleting Columns

In List mode, as you specify fields for your quick report, you may want to remove some columns to place them elsewhere. Or, you might want to delete the column from the report.

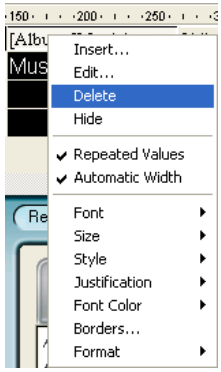
► To delete a column using the Quick Report menu bar.

1 **Select the column you want to delete.**

2 **Choose Delete from the Columns menu.**

OR

Right-click on the header of the column to be deleted, then select the Delete command from the contextual menu.



4D removes the selected column of the report.

Replacing Columns

In List mode, you can replace a column in the quick report by dragging another field over it. You can also replace a field with a formula and vice versa.

► To replace a column by a field:

1 **Drag a field from the Fields list to the column you want to replace.**
The replacement is instantaneous. If the previous field was designated as the sort level, the field that replaces it will also be substituted in the sort order list.

► To replace a column using the formula editor:

1 **Select the column you want to replace and choose Edit... from the Columns menu.**
OR
Right-click on the header of the column to be replaced, then select the Edit... command in the contextual menu.

The standard Formula editor of 4D appears, allowing you to designate the new data source of the column (field or formula).

When you print the quick report, 4D will print the results of the formula for each record that appears in the Detail row. See the "Associating Formulas with a Quick Report" paragraph on page 722 for information about adding a formula to a quick report.

Sizing Columns

By default, the Quick Report editor sizes columns automatically, as reflected in the **Automatic Width** button. It sizes each column based on the maximum length of data displayed in the column and any labels typed into the column. The Quick Report editor sizes the columns when the report is printed.
This operation is enabled for a column when the **Automatic Width** attribute has been associated with it. You can set and view the activation of this option in the **Columns** menu, in the contextual menu of the columns or using the corresponding button in the Columns toolbar:

To view the widths of each column, preview the report on the screen. See the "Executing a Quick Report" paragraph on page 748 for more information about previewing a report.

Because selecting the **Automatic Width** button or contextual menu item changes the width of a column based on the maximum width of data in the records being printed, selecting different records can change the size of the columns.

You can resize a column manually, which automatically causes the deselection of the **Automatic Width** option (where applicable). When a column is set manually, text in the column wraps within the specified area.

▶ To manually resize a column:

1 **Select the column you want to resize.**

2 **Move the pointer over the column divider in the quick report to change the pointer into a column width cursor** ⇹**.**

3 **Drag the column divider to the left or right to resize the column.**

Moving Columns

In List mode, as you specify fields for your quick report, you may want to move some columns to place them elsewhere.

▶ To move a column using the Quick Report toolbar:

1 **Select the column you want to move.**

2 **Click the move icon to move it to the right or to the left:**

OR

3 **Select Move right or Move left from the Columns menu to move the column as desired.**

4D moves the selected column.

Associating Formulas with a Quick Report

You can add a formula to a column in a quick report. For example, you can add a formula that computes employees' monthly salaries from an Annual Salary field.

► To associate a formula with a column:

1 **Insert an empty column**
OR
Click an existing column and choose Edit... from the Columns menu.
OR
Double-click an existing column's header.
OR
Right-click on an existing column header to display the Quick Report contextual menu and choose Edit.

4D displays the Formula editor, which you can use to build a formula. If you selected an existing column, the formula you create will replace the previous contents of the column.

Note Make sure that the formula you create does not change the current selection. Changing the current selection will cause problems when you print the quick report since the report is based on the current selection. For information about commands and functions that change the current selection, refer to the *4D Language Reference*.

2 **Build the formula by selecting the fields, operators, commands and/or 4D methods, then enter the desired values in the editing area.**
OR
Click the Load... button to retrieve an existing formula from disk.

For more information about the formula editor, refer to the "Global Updates" paragraph on page 979.

If you click the **Load...** button, 4D displays an open-file dialog box and asks you to select a file. When you load a file, it replaces any formula that currently appears in the Formula editor. After you load a formula, you can modify it in the editing area.

To save the formula as a file that you can retrieve and use in another column or in another report, click the **Save...** button and enter a file-name in the dialog box.

3 **Click OK to assign the formula to the column.**

4D adds a new label to the column that identifies it as a formula. You can relabel the column by typing a label into the header cell for that column. The formulas are labelled *C1* to *Cn*. These labels are the name of the variables containing the current value of the column. You can use these variables in other formulas.

Modifying the Graphic Attributes of a Report

You can modify the graphic appearance of a quick report. The Quick Report editor lets you set the following attributes:

- the character font, as well as its size, justification, style and color,
- the background color of the cells,
- the cell borders.

In addition, 4D provides a set of predefined report templates that you can modify as desired.

Note Appearance specifications are only taken into account for the "Printer" and "HTML File" output destinations (colors only). With other types of output, the formatting of reports must be carried out after the report is generated, using the receiving applications (4D View, 4D Chart or a text editor). For more information about report destinations, refer to the "Executing a Quick Report" paragraph on page 748.

Specifying Character Font, Size, Justification, Style, and Color Attributes

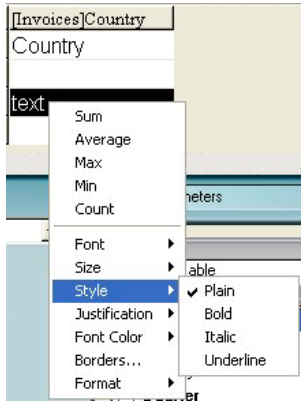
While designing your quick report, you can specify different fonts, font sizes, justification, styles and colors. You can then apply these specifications to text, data, and summary calculations within rows, columns, or cells in the quick report.

If you assign specifications to the Detail row of the report, you will not see the results until you preview or print the report.

You can specify font attributes using either the Quick Report menu commands or the Quick Report contextual menu.

► To specify a character font, font size, style, justification or color:

1 **Right-click on the row label, column header, or cell to which you want to apply the font attributes.**
A contextual menu appears.



2 **Use the Font, Size, Style, Justification or Font Color submenus to change the font attributes as desired.**
OR

1 **In the report, select the column, row or cell that you want to configure.**

2 **Choose a font in the drop-down font list and a size in the drop-down size list; then choose a style, justification and color in the Style toolbar:**

Note You can also assign a style and justification using the **Style** menu.

4D applies the parameters to the text, data or calculations included in the selected area.

Specifying the Background Color of Cells

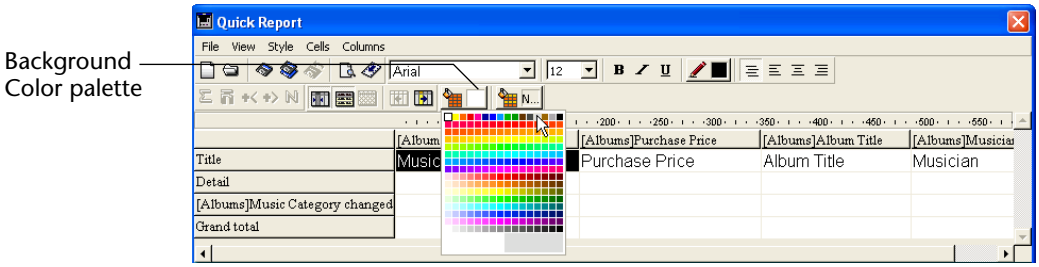
The Quick Report editor allows you to set a background color for each cell. You can set either a single color or two alternating colors, which provides better readability for tables.

It is possible to set background colors for both List and Cross-table reports.

Background colors are used with the “Print” and “HTML File” output destinations.

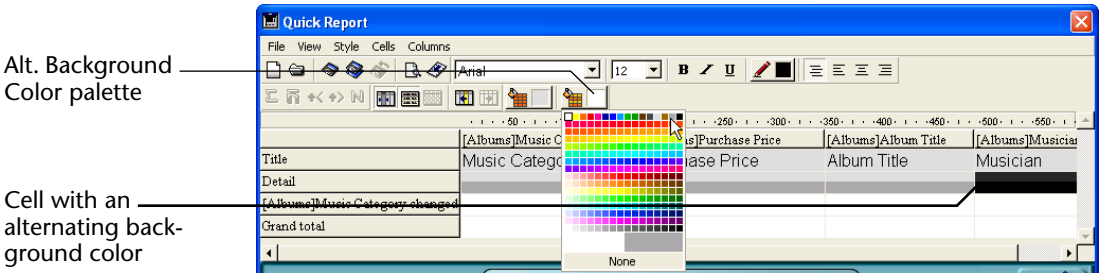
► To set a background color:

1 Select a cell, column or row, then choose the main color in the "Background Color" palette of the toolbar:



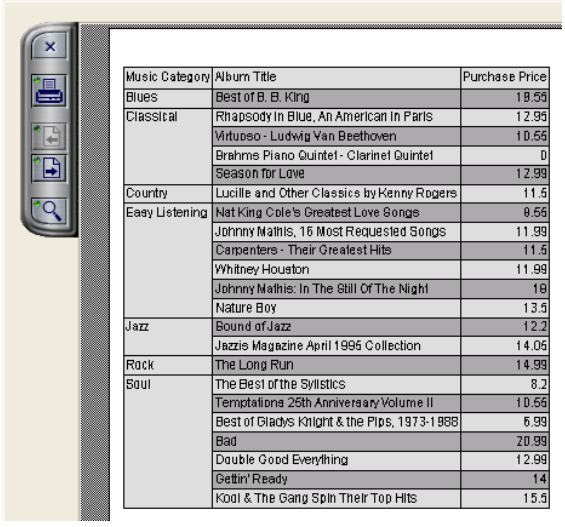
You can assign any color combination that you want to the rows, cells and columns.

2 (Optional) If you want to use an alternating background color, choose it using the "Alt. Background Color" palette:



The cells which have been assigned two alternating background colors will display both of them in the Quick Report editor.

Here is an example of a generated report in List mode (preview) that has alternating background colors:



Music Category	Album Title	Purchase Price
Blues	Best of B. B. King	19.55
Classical	Rhapsody In Blue, An American In Paris	12.95
	Virtuoso - Ludwig Van Beethoven	10.55
	Brahms Piano Quintet - Clarinet Quintet	0
	Season for Love	12.99
Country	Lucille and Other Classics by Kenny Rogers	11.5
Easy Listening	Nat King Cole's Greatest Love Songs	8.55
	Johnny Mathis, 16 Most Requested Songs	11.99
	Carpenters - Their Greatest Hits	11.5
	Whitney Houston	11.99
	Johnny Mathis: In The Still Of The Night	18
	Nature Boy	13.5
Jazz	Bound of Jazz	12.2
	Jazzis Magazine April 1995 Collection	14.05
Rock	The Long Run	14.99
Soul	The Best of the Sylistics	8.2
	Temptations 25th Anniversary Volume II	10.55
	Best of Gladys Knight & the Pips, 1973-1988	6.99
	Bad	20.99
	Double Good Everything	12.99
	Gettin' Ready	14
	Kool & The Gang Spin Their Top Hits	15.5

Background colors are also taken into account in reports generated in HTML format:

	[Albums]Album Title	[Albums]Musician	[Albums]Purchase Price
Title	Album Title	Musician	Purchase Price
Detail			
Grand total			

Album Title	Musician	Purchase Price
Rhapsody in Blue, An American in Paris	George Gershwin	12.95
Sound of Jazz	Lionel Hampton	12.2
Nat King Cole's Greatest Love Songs	Nat King Cole	8.55
The Best of the Sylistics	Stylistics, The	8.2
Johnny Mathis, 16 Most Requested Songs	Johnny Mathis	11.99
Best of B. B. King	B. B. King	19.55
Carpenters - Their Greatest Hits	Carpenters, The	11.5
Jazzis Magazine April 1995 Collection	Various	14.05
Virtuoso - Ludwig Van Beethoven	Berliner Philharmoniker	10.55
Temptations 25th Anniversary Volume II	Temptations, The	10.55
Brahms Piano Quintet - Clarinet Quintet	Benda Musicians, The	0
Best of Gladys Knight & the Pips, 1973-1988	Gladys Knight & the Pips	6.99
Bad	Michael Jackson	20.99
Double Good Everything	Smokey Robinson	12.99
Gettin' Ready	Temptations	14
The Long Run	Eagles	14.99
Kool & The Gang Spin Their Top Hits	Kool & The Gang	15.5
Lucille and Other Classics by Kenny Rogers	Kenny Rogers	11.5
Whitney Houston	Whitney Houston	11.99
Season for Love	London Symphony Orchestra	12.99
Johnny Mathis: In The Still Of The Night	Johnny Mathis	18
Nature Boy	Nat King Cole	13.5

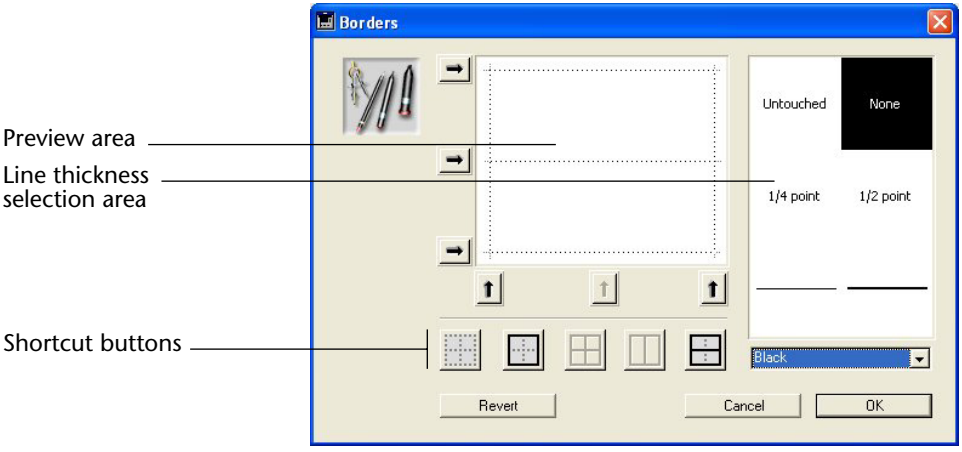
Setting Borders

You can set the borders for cells in both cross-table and list reports.

► To set the borders for a cell, a column or a row:

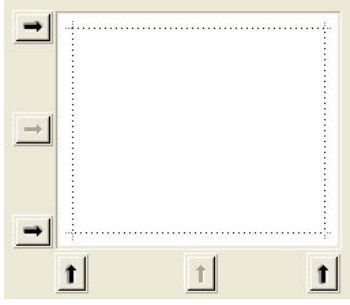
1 Highlight a cell, column or row, and either select Borders... in the contextual menu or from the Style menu.

The "Borders" dialog box is displayed:

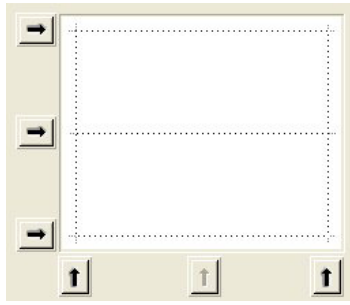


The preview area varies dynamically depending on the type of the selection: cell, column or row, but also whether a cell is in a list or cross-table type report. For example, the behavior of the center cell in a cross-table report is different from the behavior of a detail cell in a list report. The center cell is repeated both horizontally and vertically for a cross-table report, whereas the detail cell in a list report is going to be repeated only vertically. Also, other cells may not be repeated at all (titles, for instance).

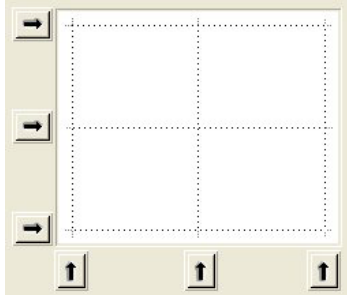
- If the selected cell is a title cell (which is not repeated), the outer lines represent the outer lines of the cell:



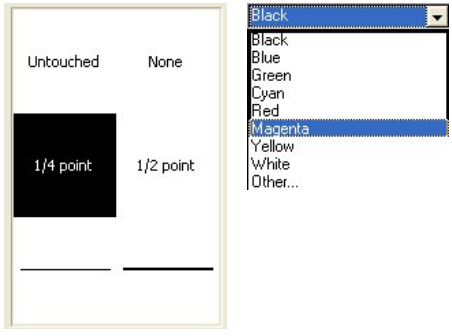
- If the selected cell is a detail cell in a List report (which is repeated vertically), the outer lines represent the outer lines of the cell repetition, and the inner horizontal line represents the border between two cells in the sequence:



- If the selected cell is the center cell in a cross-table report (which is therefore repeated both horizontally and vertically), the outer lines represent the outer lines of the repetition of cells, and the inner lines represent the borders between each cell generated from the center cell:



2 Select the line thickness and color to use:



You can use different colors for different borders.

3 **Select the borders using either the arrows or the shortcut buttons.**
You will notice that some shortcut buttons are disabled to match the type of the cell you are editing.

4 **Repeat steps 2 and 3 for each border to be configured.**

5 **Click OK to validate the changes.**
OR
Click Revert to revert to the original border settings.
OR
Click Cancel to discard the changes and close the dialog box.

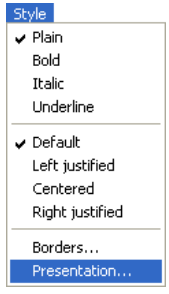
Using a Predefined Report Template

The Quick Report editor contains a set of predefined templates providing various graphic approaches for the production of printed or HTML reports. You can use all of the template as is or you can use it as a basis for constructing your own reports. A template can be applied at any time.

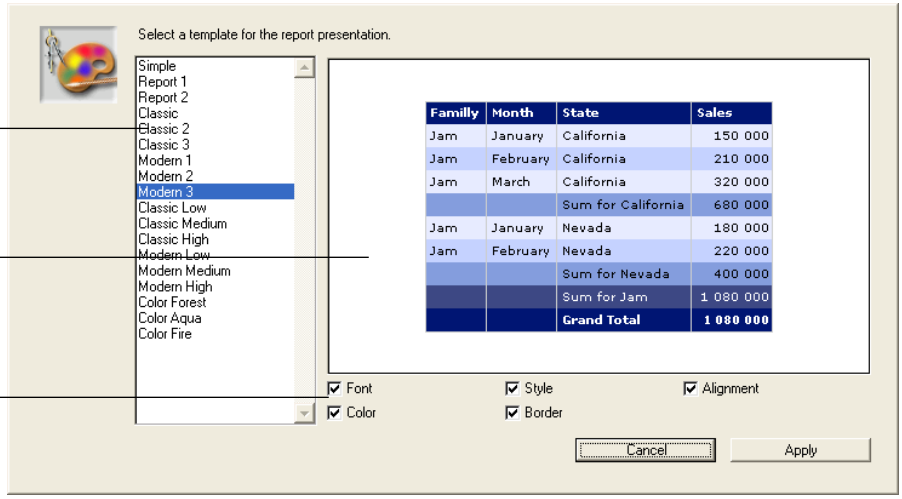
A report template combines a set of graphic characters: font, colors, style, borders and justification. You can use either all or part of the template characteristics.

► To use a predefined template:

1 **Choose the Presentation... command in the Style menu.**



The template selection dialog box appears:



2 In the list of predefined templates, click on the one that you want to use.

You can preview each template in the right-hand area of the dialog box. Choose the template that best corresponds to your needs, keeping in mind that you can always modify your report subsequently.

3 If you do not want to use all the graphic options set by the template, you can uncheck each option that you do not want to be taken into account.

By default, all the options are selected. If, for example, you uncheck the “Border” option, the borders defined by the template will not be applied to your report.

Note Only the “Color” option will be taken into account for reports whose output destination is an HTML file.

4 Click on Apply.

The selected characteristics are immediately applied to your report. If any graphic attributes were set previously, they will be replaced.

Keep in mind that certain characteristics will be only visible once the report has been generated.

Sorting Report Records

An important feature of the Quick Report editor is the ability to sort the records in your report. You sort records for two reasons:

- To view records in a particular order,
- To create groups of records and subtotal areas in the report for the purpose of reporting summary calculations for groups.

Specifying a Sort Order for a List Report

You can specify a sort order at any time; simply drag and drop a column into the Sort order list.
For example, if you wanted to sort the records of salespeople by the Sales Region field, you would drag and drop the Sales Region column into the Sort order area.

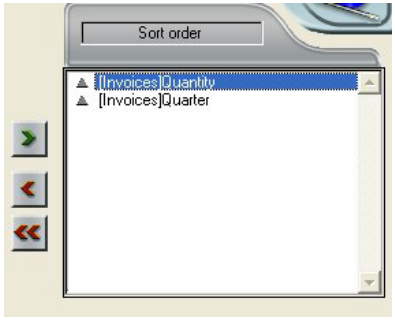
You can also sort based on a formula by selecting the column that contains the formula and placing it in the Sort order list. See the "Associating Formulas with a Quick Report" paragraph on page 722 for more information about adding formulas to quick reports.

You can sort a report on several levels. The order in which the fields and formulas appear in the Sort order list indicates the sort level.

► To specify the sort order using the field list:

1 **Drag the Field you want to set as the sort level from the Fields list to the Sort order list.**
OR
Highlight the field from the Field lists and click on the field insertion button:

The field is then added to the Sort order list.

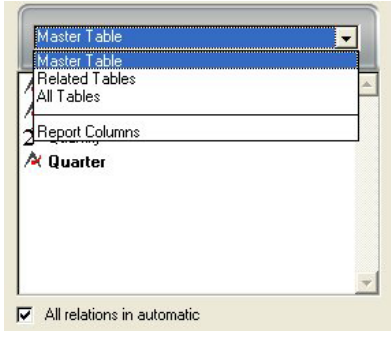


By default, sorts are carried out in increasing order (A -> Z). You can reverse the direction of the sort by clicking on the triangle next to the field in the Sort order list.

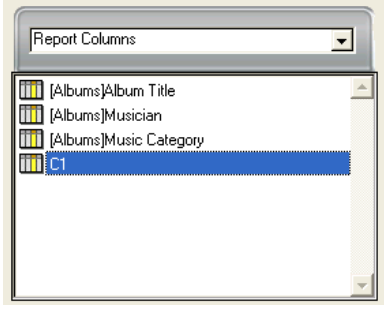
Note If the sorted field has not already been inserted into a report column, an additional column is automatically created in the Quick Report area.

► To specify a sort for a column (field or formula):

1 Select Report Columns from the table selection drop-down list.

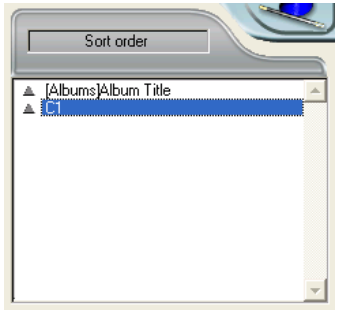


The list then displays the headers of the columns present in the report:



2 Drag the column that you want to set as the sort level and drop it in the "Sort order" list.
OR
Highlight the field from the Field lists and click on the field insertion button:

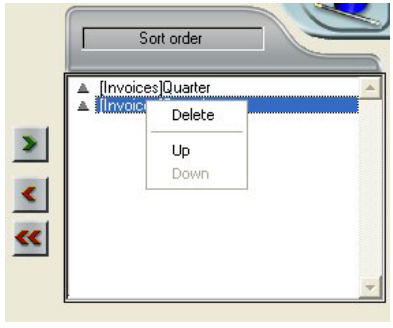
4D then displays the name of the column in the Sort order list:



Changing the Sort Level

▶ To change the level of a sort:

1 **Right-click on the field in the Sort order list in order to display the contextual menu:**



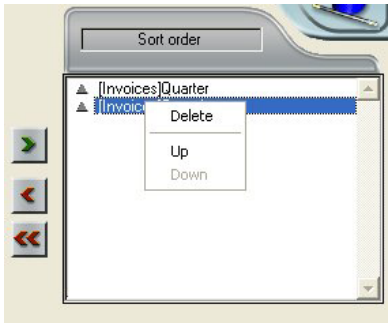
2 **Select Up or Down to move the field up or down (respectively) in the sort levels.**

Deleting a Field or Formula from the Sort List

You can delete any field or formula from the Sort list.

► To remove a field or formula from the Sort order:

1 **Select the field or formula to be deleted in the sort list and display the contextual menu using a right-click:**



2 **Select Delete.**

The level is then removed.
Note that the associated column is not removed from the report. On the other hand, the associated subtotal row disappears.

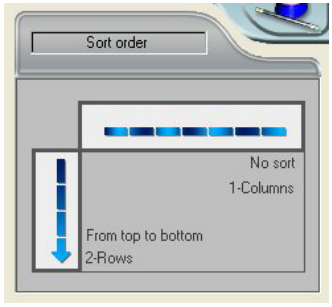
Sorting the Cross-table Values

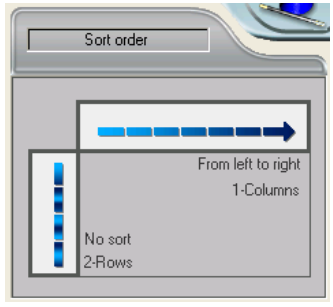
In a Cross-table report, the only values that can be sorted are the horizontal and vertical data sources (the two data sources that are used as categories in the final table).

► To sort the categories in a cross-table report:

1 **Click the sort indicators in the Sort order area.**

An arrow indicates the sort order specified:





To modify or delete the sort order of a data source, click on it again. The different possible reports appear successively.

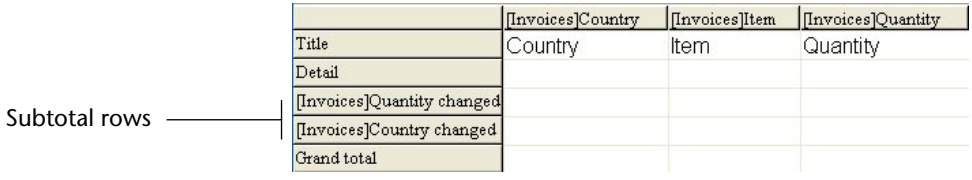
When no arrow is displayed, no sort order is selected. In this case, values will be displayed in the order they appear in the selection.

Subtotal Levels

In a quick report, you set break levels to separate or "break" records into groups according to values in one or more sort fields. A break area is printed at each break level. You can print summary calculations in the break area. The summary calculations — sum, average, minimum, maximum, count and standard deviation — are calculated for each group of records.

Break levels are determined by the sort levels and Break rows. For example, if you sort records by Sales Region, 4D inserts a break between each group of records that have the same sales region. These rows are automatically inserted when a sort is defined.

After you add a subtotal row to the quick report, you can request summary calculations on each break. For example, you can insert a summary calculation in a subtotal row to display subtotals for sales from each state in a marketing region. Refer to the "Adding Summary Calculations" paragraph on page 737 for more information about adding summary calculations to Subtotal and Total rows.



The label of a subtotal row indicates which change in value triggers the break.

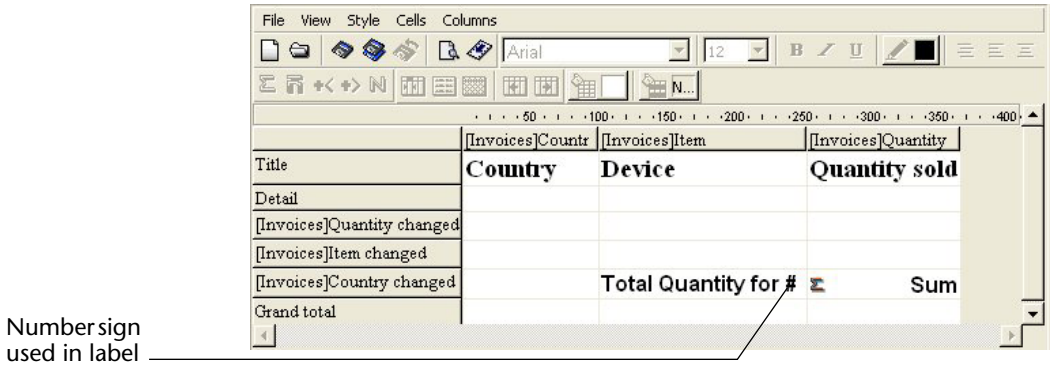
Using the Values of Break Fields in Labels

You can improve the appearance and clarity of your reports by labeling each Subtotal row using the value of the Break field.

To request that the value of a Break field be printed in a label placed in the Break area, use the number sign (#) in the label. For example, the text "Total salaries for # department" will insert the department name (in this case, the value of the Department field) in place of the number sign when the report is printed.

The number sign does not need to be placed in the same column as the Break field. It will display the value of the Break field in any cell in the Subtotal row.

The following figure illustrates the use of the number sign in a label in the Subtotal row:



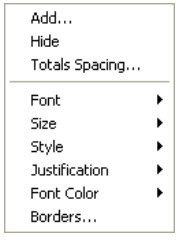
Subtotal Spacing

It is possible to configure subtotal rows in order to control the page layout and appearance of the quick report. For example, you can generate a page break after each subtotal.

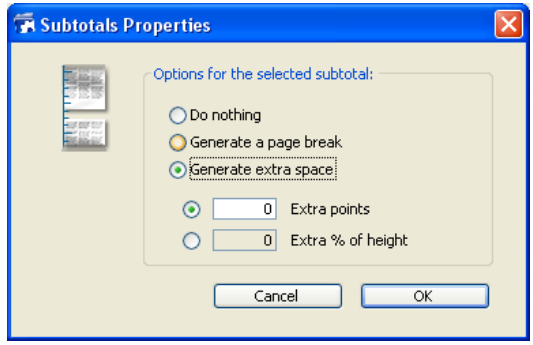
The subtotal page layout options can be used to visually set apart the different parts of the report.

► To set the spacing for a subtotal row:

1 Select a subtotal row and choose the Totals Spacing command from the File menu of the editor.
OR
Right-click on the subtotal row title and choose the Totals Spacing command:



The “Subtotals Properties” dialog box appears:



2 Choose one of the three following options:

- **Do nothing** (option selected by default): No specific property is applied to the subtotal row; it has the same spacing characteristics as the other rows of the report.
- **Generate a page break**: A page break is generated after each subtotal row in the report.
- **Generate extra space**: A specific amount of space is added below each subtotal row in the report.
  An additional option can be used to set the spacing mode:
  - **Extra points**: You set a specific height of extra space in points.
  - **Extra % of height**: You set the amount of extra space to be added as a percentage of the standard row height of the report. For example, to generate extra space corresponding to two empty rows, pass the value 200.

Adding Summary Calculations

You can add summary calculations on the contents of fields and formulas to each Subtotal row and to the Totals row. In a cross-table report, calculations can be inserted in the center row.

The calculation buttons in the toolbars of the Quick Report editor identify the summary calculation options available for quick reports.

The following summary calculations are available:

- **Sum**: Totals the values in the report or break.

- **Average**: Calculates the average of the values in the report or break.

- **Minimum**: Displays the lowest value in the report or break.

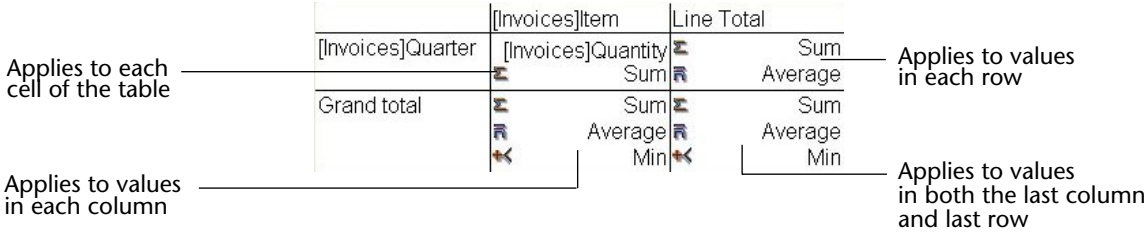
- **Maximum**: Displays the highest value in the report or break.

- **Count**: Calculates the number of records in the report or break.

- **Standard deviation**: Displays the square root of the variance of the report or break (the variance is a dispersion value around the average).

These options also appear in the Quick Report contextual menu for cells in the Subtotal and Total rows or, for a cross-table report, in the total cells and the center cell.

- List Reports
  When you place a summary calculation in the Totals row, the calculation is done for all records in the report. If you place the summary calculation in a subtotal row, separate calculations are done for the records in each break.
- Cross-table reports
  Summary calculations will apply as follows



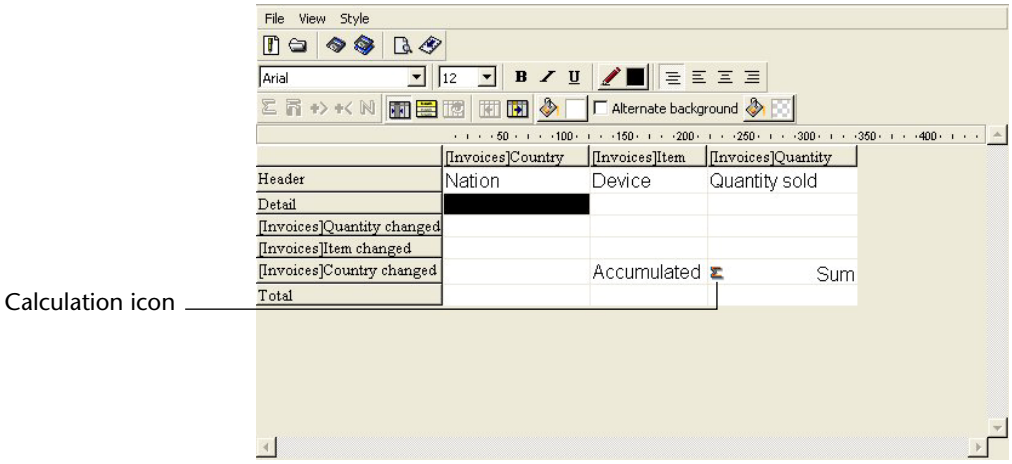
► To add a summary calculation:

1 **Select a cell where you want to insert the summary calculation.**

The Sum, Minimum, Maximum, and Average calculations work only on a numeric field or formula.

2 **Click as many summary calculation buttons as you like.**
OR
From the contextual menu, choose the desired summary calculations.

4D displays a calculation icon in the selected cell for each type of summary calculation you request. The following figure shows a Sum calculation icon in a Subtotal row:



If you add more than one summary calculation to a cell, 4D stacks the calculation icons on top of each other.

Using Calculations and Column Values in Labels

You can insert summary calculations using the following codes.

- **##S** will be replaced by the **sum** in the sub-total or total row.
- **##A** will be replaced by the **average**.
- **##C** will be replaced by the **count**.
- **##X** will be replaced by the **max**.
- **##N** will be replaced by the **min**.
- **##D** will be replaced by the **standard deviation**.
- **##xx**, where *xx* is a column number. This will be replaced by that column's value, using its formatting. If this column does not exist, then it will not be replaced

These codes can be useful when you want to mix labels and data in a cell.

Displaying Repeated Values for Break Columns

In a report with subtotals, the columns which are used to group records so that summary calculations can be done are called *Break columns*. In the report shown below, the Department field is a Break column since the records in the report are grouped by department.

When a report like this is printed, the values for the Break column are printed only once per break. In other words, a department name is printed only for the first record in the group and is not repeated until the department changes.

First Name	Last Name	Department Name	Salary
Biff	Davis	Accounting	43780
Smeldorf	Garbando		19610
Alan	Hull		41460
Bryan	Pfaff		26440
Shirley	Ransome		36040
Marlys	Wilson		36500
		Sum for Department : Accounting	203830
Kathy	Forbes	Engineering	18840
Dennis	Hanson		40520
Mary	Smith		55000
Andy	Venable		43520
Lance	Wolfram		27300
		Sum for Department : Engineering	185180

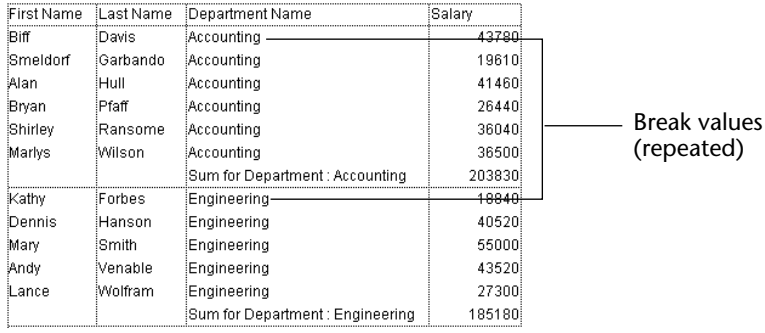
Non-repeating break values

In some cases, you may want to repeat the values for the Break columns so that they appear for every record in the Break area. You do so by selecting the **Repeated Values** column property. This can be done either by clicking the **Repeated Values** button in the toolbars  , by choosing the **Repeated Values** menu command in the Quick Reports contextual menu for that column, or by selecting Repeated Values from the Columns menu.

► To display repeated values for fields in a column:

1 Select the column by clicking the header row for that column and choosing the Repeated Values command in the Columns menu or clicking the Repeated Values button  in the "Columns" toolbar.
OR
From the contextual menu for that column, choose Repeated Values.

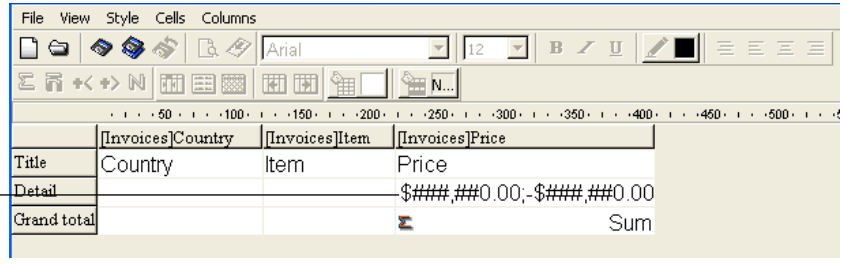
The following figure shows the previous report after the **Repeated Values** check box has been checked for the Department Name column.



Setting Display Formats

You can specify display formats for columns that contain numeric, alphanumeric, date, time and picture data.

For example, if you are displaying prices in a column, you can add a numeric format to the Detail cell for the Price field. The $###,##0.00;-$###,##0.00 format places a dollar sign ($) to the left of the number and can display dollar amounts from -$999,999.99 to $999,999.99:



If your report includes Alpha fields such as a telephone number or Social Security number, you can use an Alpha format. If your report displays dates, times or pictures, you can also assign display formats to them.

4D provides different default display formats. However, you can create your own display styles for numeric and alphanumeric formats. For more information about this point, refer to the "Display Formats" paragraph on page 521.

Assigning the Display Format

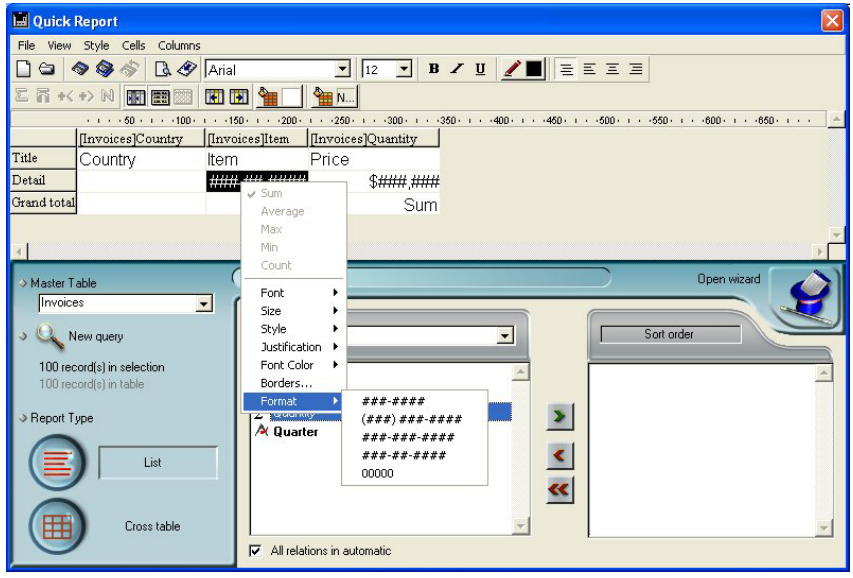
You can associate a display format with a cell by selecting it in the contextual menu or by entering it directly in the cell.

► To assign a display format to a numeric, alpha, date, time or picture field:

1 **Right-click on a cell of the column with which you want to associate a style.**

The contextual menu of the editor appears.

2 **Select the desired display format from the Format submenu:**



The contents of the submenu vary according to the type of data contained in the column: numeric, alphanumeric, date, time or picture. If the report column contains data that cannot be formatted, the Format command does not appear in the contextual menu.

The format is entered in the Detail cell. If you have also requested summary calculations for the column, the format specified in the Detail cell will automatically be applied to the summary calculations. The only exception is the “Count” calculation which is always displayed as an integer and does not accept any formatting symbols such as the dollar sign.

You can also enter the formats to be applied manually by clicking twice in the cell and entering the elements of the format using the keyboard.

Different formats can be applied to different columns in the report.

Hiding and Showing Rows and Columns

4D lets you hide rows or columns in a quick report in List mode. If desired, you can show a hidden column or row.

Hiding rows is useful when you want the report to include only summary calculations. For example, hide the Detail row if you want to display only the summary calculations that appear in the Totals and Subtotal rows. You can also use this feature to hide a Subtotal row or the Totals row.

You can hide a column if you need to use the column as a sort column, but do not want the report to display it.

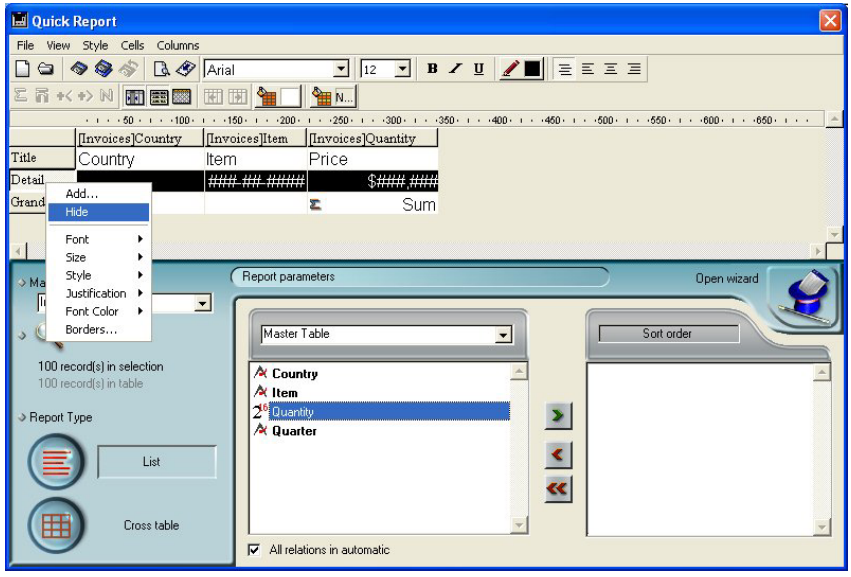
You can hide/display a row or column using either the Quick Report contextual menu, the Columns menu or the “Columns” toolbar.

Note You cannot hide a row or column in a cross-table report.

► To hide a row or column:

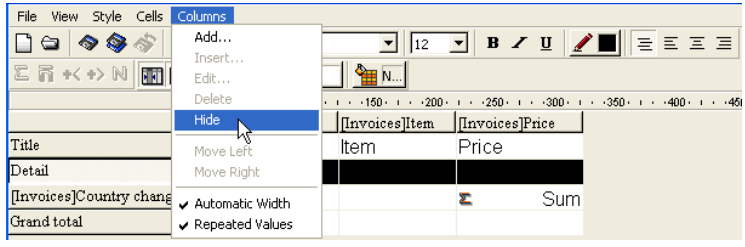
1 Select the row or column you want to hide by clicking on its header.

2 Right-click on the header of the row or column to be hidden, then choose Hide in the contextual menu:



OR

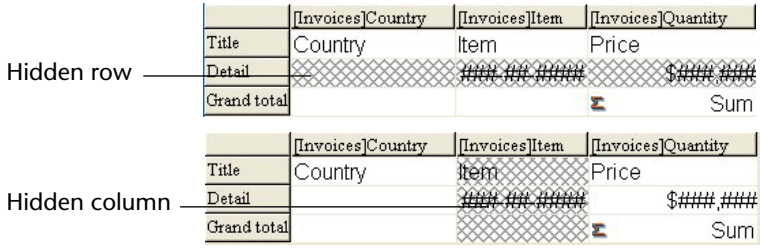
Choose the Hide command in the Columns menu of the Quick Report editor.



OR

Click on the button of the "Columns" toolbar.

4D displays the column in gray to remind you that the row will not appear when you print or preview the quick report.



Showing a Hidden Row or Column

When a row or column is hidden, a check mark (√) is displayed next to the **Hide** menu command in the **Columns** or contextual menu. In addition, the corresponding button of the "Columns" toolbar is pressed in.

You can display a hidden row or column by choosing **Hide** again from either the **Columns** or contextual menu, or by pressing the corresponding button in the "Columns" toolbar again. When you do so, the row or column is displayed normally in the Quick Report area.

Adding Page Headers and Footers

Before printing a quick report, you can add page headers and footers. You specify page headers and footers in the Headers and Footers dialog box. Use this dialog box to do the following:

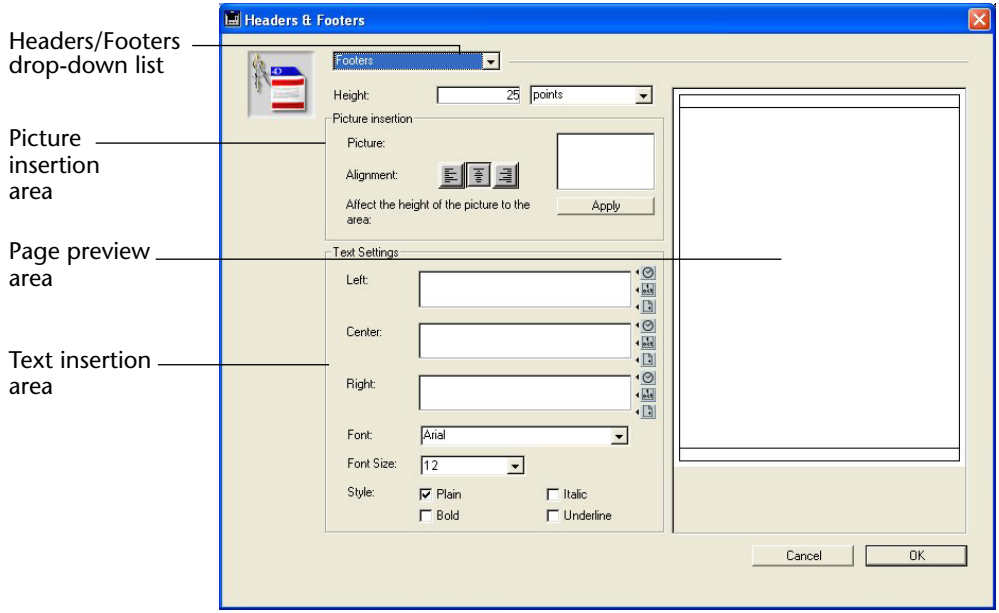
- Add page header and footer text or pictures,
- Specify the size of the page header and footer areas,
- Use separate text for left, center and right parts of the header and footer,
- Specify fonts, font sizes, and font styles for page header and footer text,
- Insert codes that add page numbers and the date and time to your reports.

You can only specify page headers and footers when printing to a printer. However, once they are defined, they are kept with the report, even if the destination is modified. See the "Executing a Quick Report" paragraph on page 748 for more information about alternate output destinations.

► To add page headers and footers:

1 Choose Header and Footer... from the File menu.

The Headers and Footers dialog box appears.



Note The preview area takes the print format configuration into account.

The Headers and Footers dialog box lets you specify both headers and footers using the same screen. Use the Headers/Footers drop-down list to indicate the one you currently want to define.

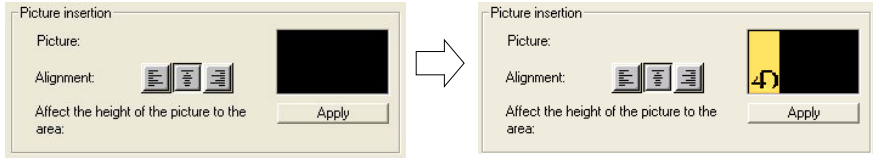
2 Choose Headers or Footers from the selection menu.

3 Enter the header or footer height in the Height area.

When you first enter the dialog box, the header and footer heights are set to 25 points each. You can not only change these values, but can change the measurement scale to enter values in inches or centimeters as well.

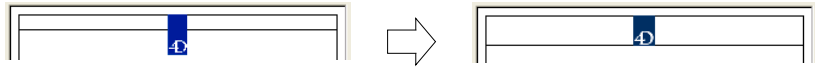
As you enter the header and footer height, the lines on the page preview area change to indicate the size of the header and footer as they will appear in the printed report.

4 **If you want to use a picture, paste it in the picture area (using standard copy-paste commands):**

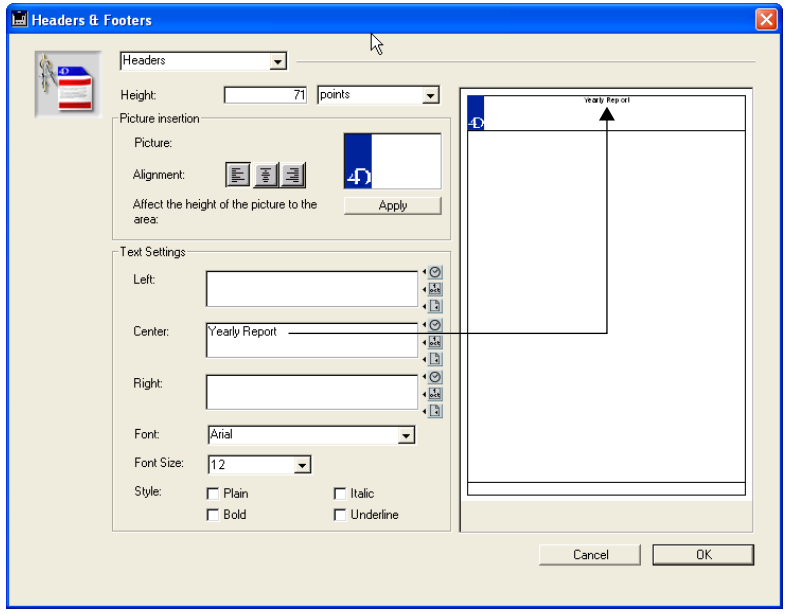


5 **Select its alignment by clicking on the alignment icons:**

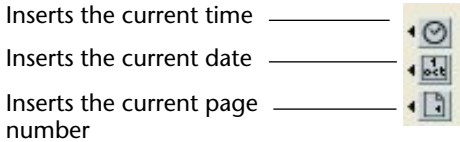
6 **If you want the height of the picture to become the height of the header/footer, click the Apply button:**



7 **Select an entry area and type the header or footer text.**



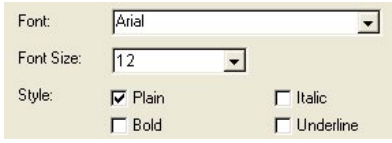
To the right of each entry area, there are three buttons that let you enter variables into the entry area.



You can insert the current page number, time of execution, or date of execution.

You can also use the code for the variables directly: #H for the time, #D for the date and #P for the page number.

8 **Assign the font attributes for the header/footer.**



9 **Click OK to validate the changes.**

Executing a Quick Report

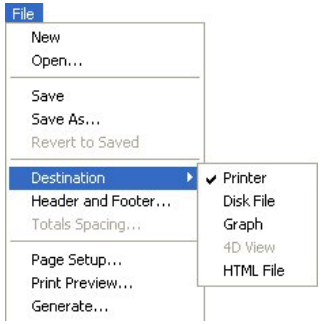
After you have completed your report design, you can "print" the quick report. You can print a quick report to a variety of output types:

- On the standard printer selected,
- To disk, in a Text file,
- To a graph,
- To a 4D View document,
- To an HTML document.

► To select an output destination:

- **Choose Destination from the File menu.**

The Destination submenu contains five items.



Printer

This option uses the printer you have chosen in your current print settings. If you are printing to a printer, you can preview the report before printing it.

▶ To print to a printer:

1 **In the File menu, choose Printer from the Destination submenu.**
This option is selected by default.

2 **If necessary, select Page Setup from the File menu and specify your print parameters.**

Note This command is only available when the **Printer** destination is selected.

3 **If you want to preview your report, select Print Preview... in the File menu.**

Note This command is only available when the **Printer** destination is selected.

The print preview dialog box appears, containing the report that you have defined with the current data of the database.

4 **Select Generate... in the File menu.**

5 **Choose the settings that are appropriate for your report and click the OK button.**

Disk File

This option sends your quick report to a disk file that you can open and modify with other applications, including text editors and spreadsheets. This option exports the records in the quick report to a text file.

When you use this option, 4D automatically uses the column headings as the first "record" that is exported.

► To generate the report in a Text file:

1 **In the File menu, choose Disk File from the Destination submenu.**

2 **Choose Generate... from the File menu.**

4D displays a standard create-file dialog box and asks you to enter a filename.

3 **Enter a filename and click OK.**

4D displays a dialog box that keeps you informed of the progress of the operation.

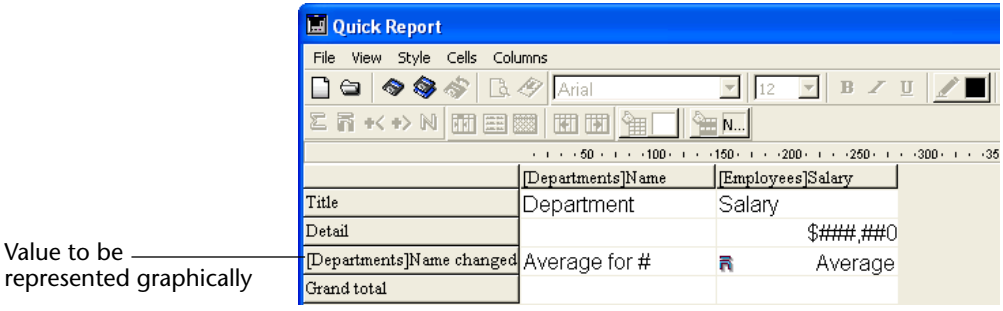
After the report is printed to a file, 4D returns you to the Quick Report editor. Remember to change the output device if you want to resume sending a quick report to a standard printer.

Print to Graph

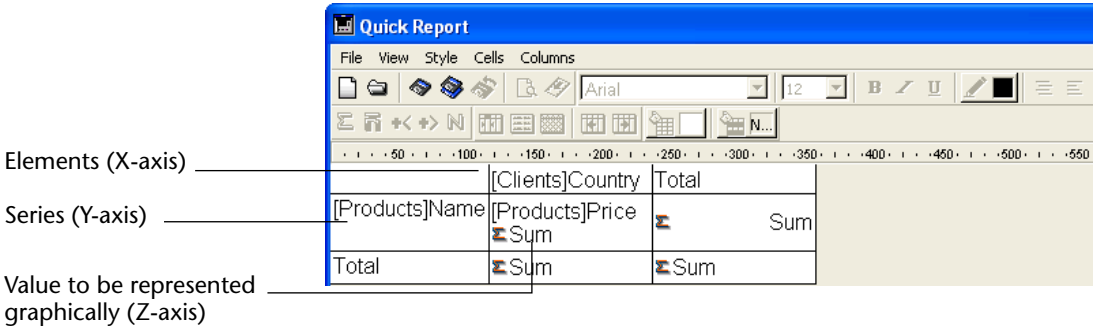
This option directs the report to 4D Chart, 4D's plug-in for plotting data. When you choose the Graph item in the Destination submenu, your report is presented as a graph rather than in table form. Your graph can then be printed using 4D Chart.

- In **list mode**, 4D Chart uses only the summary calculations and labels in the Subtotal row. It uses the leftmost non-numeric column for the Categories axis (the horizontal axis). To use the Graph feature, your report should:
  - Include from one to five numeric fields or formulas. These columns will be assigned to the values axis in the graph.
  - Use one type of summary calculation per numeric field.

For example, if you want to graph the average salary by department, you should create a quick report with two columns, Department Name and Salary as well as a Subtotal row consisting of the Average summary calculation for Salary:



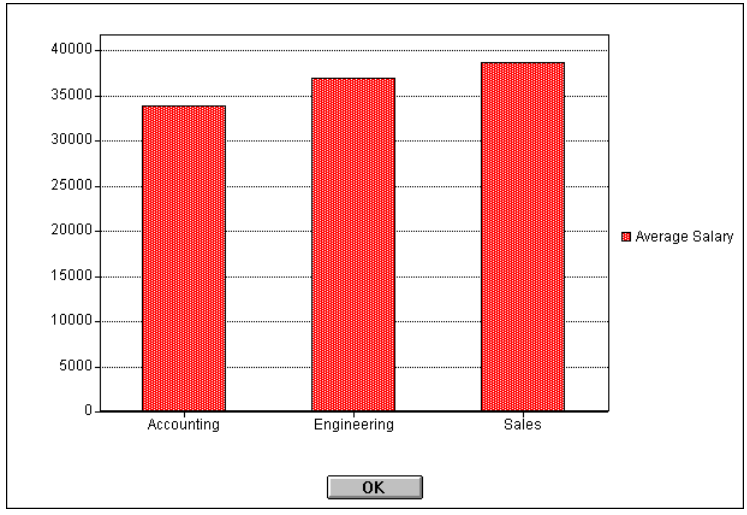
- In **cross-table mode**, 4D Chart uses two data sources and the value cell. The last row and column are ignored.
  For example, if you want to show the amount of sales per country for a set of products, you can define the following report:



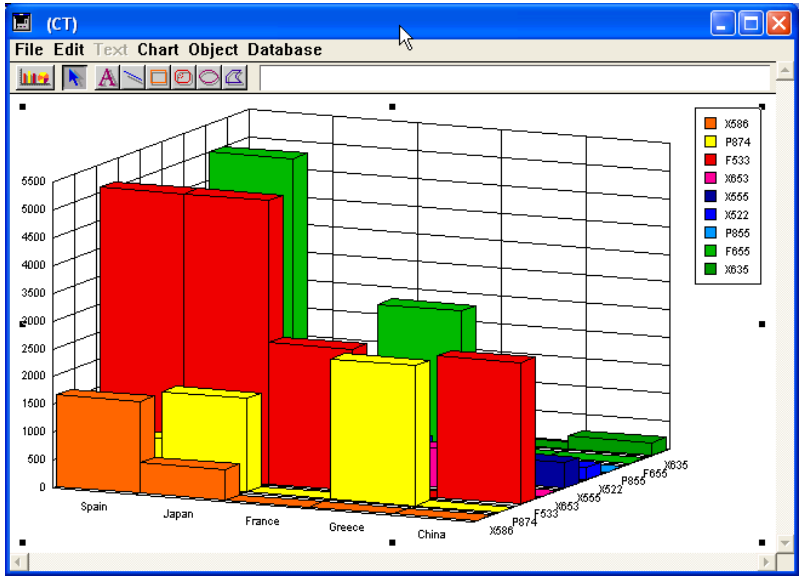
► To print to a graph:

1. **In the File menu, choose Graph from the Destination submenu.**
2. **Choose Generate... from the File menu.**

- For list reports, 4D graphs your data as a 2D column graph.



- For cross-table reports, 4D graphs the data as a 3D column graph:



You can use 4D Chart to select another graph type or print the graph. For more information about 4D Chart, please refer to the chapter “Graphs” on page 1053.

Printing to a 4D View Document

This option directs the report to 4D View, 4D's plug-in for displaying data. When you choose the 4D View item in the Destination submenu, your report is presented as a 4D View window rather than in table form. 4D View uses all the data in the report.

Note The **4D View** destination is only available if this plug-in is installed in your 4D environment, and if you have the appropriate licenses.

▶ To generate a report in a 4D View document:

1 **In the File menu, choose 4D View from the Destination submenu.**

2 **Choose Generate... from the File menu.**

4D creates a 4D View window containing the data. You can modify its contents, print it or save it using the 4D View menu commands.

Printing to an HTML Document

This option sends your quick report to an HTML file. When you choose this option, it uses the default HTML template unless it was changed programmatically.

When you use this option, 4D automatically uses the column headings as the first "record" that is exported.

▶ To print to an HTML file:

1 **In the File menu, choose HTML File from the Destination submenu.**

2 **Choose Generate... from the File menu.**

4D displays a standard create-file dialog box and asks you to enter a filename.

3 **Enter a filename and click OK.**

4D displays a dialog box that keeps you informed of the progress of the operation.

After the report is printed to an HTML file, 4D returns you to the Quick Report editor. Remember to change the output device if you want to resume sending a quick report to a standard printer.

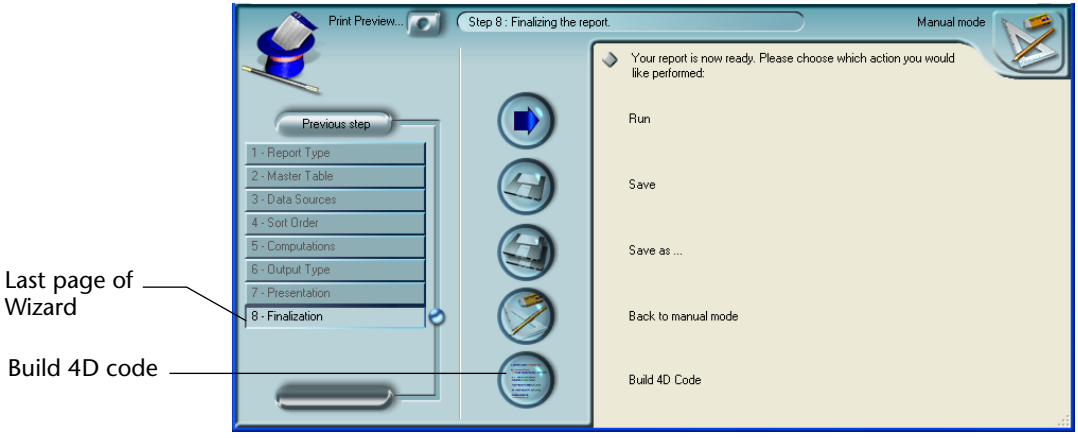
Note The character set used for generating HTML code is specified using the "Standard Set" parameter of the application Preferences (see the "Options Page" paragraph on page 225).

Generating 4D Code

The Quick Report editor allows you to automatically generate the 4D code corresponding to a defined report and paste it in the Clipboard. This function allows manual building of quick report outlines that can be customized using 4D language.

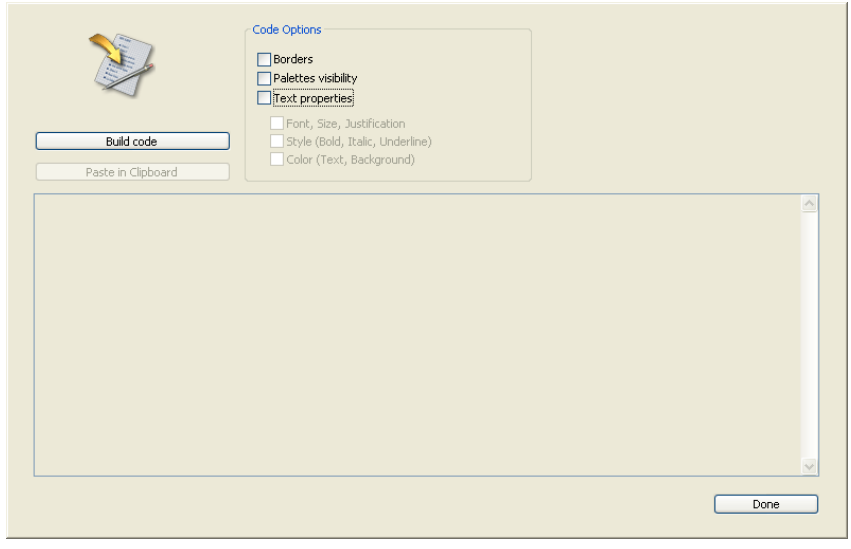
This feature is only available in Design mode.

The button for code generation is located on the last page of the Quick Report Wizard:



Even if you built the quick report manually, you can use this feature by displaying the “Finalization” step of the wizard.

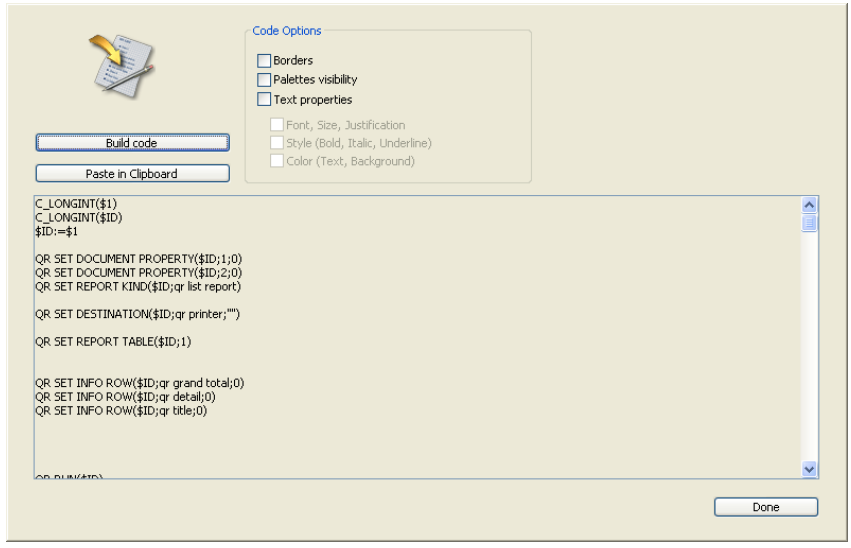
When you click on this button, a dialog box appears that allows you to set the parameters related to code generation:



The following options are available:

- **Borders**: Check this option to specify that you want to include the cell border properties in the generated 4D code.
- **Palettes visibility**: Check this option to specify that you want to include the current palettes display status (Standard, Styles, Columns...) in the generated 4D code.
  It is advisable to deselect this option if the code is intended for use in an offscreen area.
- **Text properties**: Use this option to specify that you want to include the font style definition of the report (font, style, etc.) in the 4D code.
  If this option is not checked, the three following options have no effect.
  - **Font, Size, Justification**: When this option is checked, the generated 4D code includes information regarding font, font size and font justification in the quick report template.
  - **Style (Bold, Italic, Underline)**: When this option is checked, the generated 4D code includes information regarding text styles in the quick report template.
  - **Color (Text, Background)**: When this option is checked, the generated 4D code includes information regarding the text colors in the quick report template.

  Once you have defined the options, click on the **Build code** button to generate the 4D code; the code is then displayed in the Preview area of the dialog box:



You can modify the code options and click again on the **Build code** button in order to update the preview area.

If you are satisfied with the code generated, click on the **Paste in Clipboard** button, then click on the **Close** button to close the dialog box.

You can then paste the code in a method or a text file using the standard shortcut **Ctrl+V** (Windows) or **Command+V** (Mac OS), or the **Paste** command in the **Edit** menu.

11 Methods

You can attach a method to a 4D object to specify the object's action. A method is a series of instructions that tell the object to do something. For example, you can use methods to:

- Enforce business rules during data entry,
- Calculate values for fields and variables,
- Manage interface elements such as combo boxes, hierarchical lists, and tab controls,
- Manage drag-and-drop actions,
- Assign actions to custom menu commands,
- Create and manage multiple processes,
- Manage transactions,
- Manage custom reports,
- Regulate multi-user database access.

This chapter provides information about using 4D's Method editor to create and modify methods. To learn more about 4D's programming language, refer to the *4D Language Reference* manual, which provides detailed information about the programming commands and syntax.

Overview

Types of 4D Methods

You can create the following five types of methods:

- **Object methods**: Object methods are associated with individual objects on a form, such as fields, buttons, drop-down lists, and tab controls. They can be used for such purposes as assigning initial values, managing and validating data entry, or managing drag-and-drop actions.
- **Form methods**: Form methods are attached to individual forms (table forms or project forms). A Form method can manage everything that happens when a form is used for data entry, screen display, or printing. Alternatively, you can use Object methods to manage individual objects on the form. Form methods are generally used to control the interaction between different objects and to manage the form as a whole. A typical use of form methods is calculating values based on more than one field. Since the calculation needs to be made each time the values concerned are modified, this type of calculation is placed in a form method.
- **Triggers**: Triggers are run when specific events occur at the database engine level. The four events that can launch triggers are related to record management: loading a record, saving a new record, saving an existing record or deleting a record.
  For more information about triggers, please refer to the "Triggers" section of the 4D *Language Reference* manual.
- **Database methods**: Database methods run automatically when certain worksession-related events occur. 4D provides blank methods corresponding to these events. You can add code to any of these methods.
  For more information about database methods, please refer to the section "Database Methods" chapter of the 4D *Language Reference* manual.
- **Project methods**: Project methods can be called by other methods anywhere in the database, by users, by the Web server or by custom menu items. You can create as many project methods as you want.

Introduction to Using Methods

A 4D method is a series of instructions that causes 4D to perform an action or a series of actions. For example, the following project method allows users to add records to a table:

```
INPUT FORM ([Customers];"Input")
Repeat
   ADD RECORD ([Customers])
Until (OK=0)
```

This method would be attached to a menu item in a custom application. When the user chooses this menu item, 4D runs the method. It makes the [Customers]Input form the current Input form and presents it to the user as a blank form, ready for data entry.

The user can continue to enter new record until he or she presses the Cancel button on the form. During data entry, any table, form, or object methods will run when the appropriate events occur.

4D methods are created with components of a *procedural language*. The following are the main elements of the language:

- **Fields**: You can use fields from any table. For example, a method can use a value that is stored in a field, or it can change that value and store a new value in the field.
- **Object names**: You can use the names of objects on a form. For example, you can resize an object, change its color, enable or disable buttons, or modify the font, font size, or style.
- **Variables**: You can temporarily store a value in a variable and use it later in the same method or in a different one. You can create a variable in any method, and you can use or change its value in any method.
- **Pointers**: Pointers let you write generic code that doesn't refer to database objects by name. Instead, a pointer to each object is used. Each time the generic code is used, you can "point" to different database objects. For example, if you substitute a pointer to a table for the table name "[Customers]" in the previous example, you could then reuse the code for any table. For more information on pointers, see the chapter "Arrays and Pointers" in the *4D Language Reference* manual.
- **Operators**: You can use symbols to instruct 4D to carry out an operation such as multiplication, addition, and so on.

- **Commands**: You can use commands in the language to instruct 4D to perform an action. For example, the ALERT command displays a message in an alert dialog box. The NEXT PAGE command displays the next page of a multi-page form and the ORDER BY command sorts the records in the current selection.
- **Functions**: You can use functions in the language to calculate values. For example, you can calculate an average of several values with the Average function. You can calculate a subtotal for a report with the Subtotal function.
- **Constants**: Constants are expressions whose values are invariable. Their purpose is to simplify and clarify the code. For example, it is more practical to write SET COLOR(MyObject;Grey) than SET COLOR(MyObject;14).
- **Flow of control**: You can control when code executes with flow of control structures. The 4D language includes the following control structures:
  - If…Else…End If
  - Case of…Else…End Case
  - While…End While
  - Repeat…Until
  - For…End For

  You use logical tests in these structures to determine whether or how many times the code executes. All of these elements are discussed in detail in the 4D *Language Reference* manual.

Note It is also possible to use SQL in 4D methods. This standard language is used in particular for creating, managing and querying databases. In the same method, you can use either the 4D language, SQL or both. For more information about SQL in 4D, please refer to the *4D SQL Reference* manual.

Statements

A method is composed of statements, each statement consists of one line in the method. A statement is an instruction for 4D to carry out. For example, the following line is a statement:

[People]Start Day:= **Current date**

This statement places the current date in the Start Day field of the [People] table. Current date is a function that returns the date based on the system date. [People]Start Day is a field.

Notice that the statement specifies the table name, surrounded by square brackets, with the field. When writing project methods, you specify the table name to which a field belongs to avoid possible confusion with other fields with the same name. However, when in a form or object method, you can specify fields from the form's table without specifying the table name. Table names are written within square brackets.

The previous statement is typical of statements that calculate or work with values. It starts with the field in which the value is to be placed and uses the assignment operator to point at the calculation that determines the value. The calculation is performed by whatever follows the assignment operator. The assignment operator is a colon and equals sign (:=).

You use the assignment operator whenever you need to store a value in a field, an object, or a variable. It takes the following form:

Field/Object/Variable := *Calculation*

The value container is the field, object, or variable in which you want to place the value. The calculation is the operation that results in the value you want to store. The assignment operator assigns the value that is calculated on its right into the container on its left. You will see several examples of this in the next few paragraphs.

A statement may be simple or complex. Although a statement is always one line, that one line can be as long as needed (up to 32,000 characters).

The following method displays the third page in a multi-page form:

GOTO PAGE (3)

When you use the GOTO PAGE command, you instruct 4D to display the page indicated in the parentheses.

Notice that the command is in bold capital letters; this is the way that 4D displays commands in the Method editor by default. This convention is used in all examples in the 4D documentation. You can change the style of each type of object in the application Preferences.

You do not have to type commands in all caps; 4D automatically changes the display.

Most commands require additional information to carry out the instruction. This additional information is called an argument to the command. An argument contains data that a command needs in order to complete its task. In this case, the GOTO PAGE command needs the page number to go to. An argument is always placed within parentheses following a command.

Execution of a Method

The execution of statements in a method always follows a strict order. When the method is executed, it begins at the first line and works its way down to the last line (Obviously, the order of execution is determined by the flow of control structures in the method).

This section examines a multi-line method in detail in order to establish some of the terminology, concepts, and common aspects of methods.

The following method computes the total amount due on an invoice:

```
vSales Tax:= Total Purchases * Tax Rate
vTotal:= Total Purchases + vSales Tax
[Report]Total Due:= Round (Total; 2)
```

This method is attached to an object that will be printed on an invoice. As you follow the method line-by-line, you will see that the later lines depend on previous lines. Because of the strict order in which statements are executed, you can depend on a value being available when it is needed.

Here is the first line of the method. It calculates the sales tax for the purchase:

```
vSales Tax:= Total Purchases * Tax Rate
```

In this method, “vSales Tax” is a variable. You can create a variable at any time by typing its name. You name the variable on the left of the assignment operator and calculate its value on the right. Subsequently, whenever you need that value, you can simply use the variable name. A variable can be created at any time in a method. It can be assigned a value, as here, which can then be used by a later statement.

The variable "vSales Tax" is a *process variable*. A process variable is a variable which works within the current process. In 4D, everything occurs within a process, even if the process is one created by 4D. (You will learn about processes in the chapter "Processes" on page 929.)

When a statement creates a process variable such as vSales Tax, a portion of memory is set aside and assigned the name "vSalesTax." Since it is a process variable, any method in the current process can use the value in vSalesTax. The variable remains in memory until the process is closed. *Local variables* temporarily store values that can be used within the method, but which cannot be used by any other method.

Variables follow the same naming rules as fields. Typically, you would use a consistent convention so that you know that you are using a variable. In the previous example, the variable is indicated by a lowercase v as the first letter in the vSales Tax variable. For more information about variables and the different types of variables, refer to the 4D *Language Reference* manual.

For the vSales Tax variable, the assignment operator (:=) assigns it the result of a calculation. In this case we would say, "vSales Tax *gets* Total Purchases times Tax Rate." The Total Purchases field contains the total amount of purchases for the current invoice and the Tax Rate field contains the tax rate to be used to calculate the tax. (A Tax Rate field could contain different rates based on the address of the customer.)

When an object method uses field names, the instruction is to use the value in that field on the current record. When another record is displayed, used, or printed, the method is executed again, using values in the new record.

The first line of our example method multiplies the values in the Tax Rate and Total Purchases fields and stores the result in the variable vSales Tax for each record that is used (a value entered or an invoice printed).

```
vSales Tax:= Total Purchases * Tax Rate
```

Here is the second line of the method. It creates a variable that contains the grand total of purchases plus sales tax:

```
vTotal:= Total Purchases + vSales Tax
```

The first component of the statement above is the vTotal variable. It is assigned the value generated by the addition of Total Purchases (a field) and vSales Tax (a variable). It does not matter to 4D whether a value is stored in a field or in a variable. As long as the field and the variable store data of the same type, the addition operator (+) simply adds the two values together.

Notice that this statement uses the vSalesTax variable that was created in the previous line. It also creates the vTotal variable which will be used in the following line.

Notice also that the value of Total Purchases has been used twice: first in the previous line to calculate the vSales Tax variable, and second in this line to calculate the vTotal variable. Nothing happened to the value in the field in either case. In each of these two lines, the assignment operator placed a value in a variable. These statements use the value in the Total Purchases field, but do not change that value.

The assignment operator (:=) places a value in the field, object, or variable to its left. Nothing will happen to fields, objects, or variables to the right of the assignment operator.

Here is the third line of the method. It stores a value in a field in a different table:

[Report]Total Due:= **Round** (vTotal; 2)

The first component is a field. Notice that the table name has been specified because it is different from that of the master table. Table names are always placed in square brackets [like this]. If you have to use parentheses as well, you would use both ([like this]).

The calculation is performed by the Round function. The Round function rounds off a value to a specified number of places.

Most functions require additional information to carry out the calculation. Additional information for a function is the argument to the function. A function may have several arguments. In this case, the Round function needs two arguments: the number to round off and another number specifying how many places to round off to. Here, the number to round off is the value of the vTotal variable, and the number of places to round off to is 2. The two arguments are separated by a semicolon. The variable vTotal and the number 2 are *arguments* to the Round function.

Notice that the function's name has only the first letter capitalized. This is the naming convention used for 4D functions. As with commands, you do not need to enter the capital letter. When you press the **Enter** key or click on another line in the method, 4D automatically capitalizes the first letter of the function and makes it bold (a convention which can be modified in the Preferences).

Where to Put an Object Method

You can attach an object method to any field or other active object. An object method can refer to values in other fields and objects. The general rule is to attach an object method to the active field or object, the one that receives the data entry or the one that is clicked or otherwise activated. An object method that is to be executed when a button is clicked should be attached to the button object. An object method that capitalizes entries in a field should be attached to the field.

However, if you placed the method

```
Grand Total:= Total + Sales Tax
```

in the Grand Total field, the method would not function properly since you would have to type something in the Grand Total field in order for the method to be executed.

You need the statement to be executed whenever the values in the Total or Sales Tax fields change. In order for the calculation to take place, you should place the statement in the form method or use it in object methods for both the Total and Sales Tax fields.

Managing Methods

Creating or Opening Methods

Methods are created using the Method editor. The Method editor provides you with tools to create, test, and edit any type of method.

This section describes the different ways of creating and opening methods (object methods, project methods, triggers and form methods).
Keep in mind that blank database methods are already created in all the applications; you can just open them from the Explorer.

4D Server Object locking occurs when two or more users attempt to modify the same method at the same time. If a user opens a method in the Design environment, the method is locked. Other users cannot modify this same method until the first user frees it by closing the window. In the meantime, the method can be opened in read-only in order to copy all or part of its elements.

Creating an Object Method

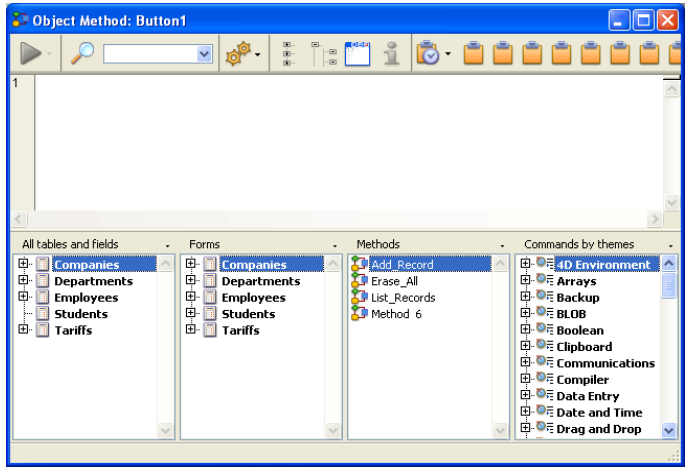
Object methods are created for an object on a form. You start in the Form editor, with a form displayed on the screen.

► To create or open an object method in the Form editor:

1 **In the Form editor, hold down the Alt key (Windows) or the Option key (Mac OS) and click the object.**
OR
Select the object, then choose Object Method from the Object menu.
OR

Click the object using the right mouse button and choose Object Method from the contextual menu.
OR
In the Property List, click the object method Edit... button.

A window of the Method editor appears, blank if you have just created the method.



The Method editor is described in the section "Method Editor Window" on page 781.

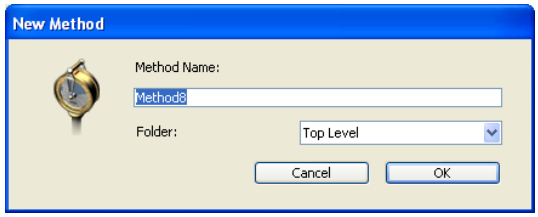
Creating or Opening a Project Method

You can create or open a project method using the **File** menu, the Methods page of the Explorer, the Menu Bar editor or a window of the Method editor.

► To create a project method from the **File** menu:

1 **Choose New > Method... in the File menu or using the New button of the 4D toolbar.**

4D displays the New Method dialog box.



2 **Enter a method name.**

Method names can be up to 31 characters long. They can include letters, numbers, space characters, and underline characters.

3 **(Optional) Select a folder for storing the method.**

If you select a folder name, the method will be placed in this folder. Folders can be used to organize the objects of your application and are managed on the Home page of the Explorer. For more information, refer to the "Home Page" paragraph on page 103. By default, the method is created at the Top Level, i.e., outside any storage folder.

4 **Click the OK button.**

4D opens an blank Method editor window where you can begin writing the new method.

▶ To create a project method from the Explorer:

1 **Display the Methods page of the Explorer.**

The hierarchical list of project methods, database methods, triggers and form methods appears.

2 **Highlight the Project Methods item.**

3 **Click the add button below the list.**

The dialog box where you can name the method and assign it to a folder appears.

4 **Enter the method name and click OK.**

▶ To open a project method from the **File** menu:

1 **Choose Open > Method... in the File menu.**

The Methods page of the Explorer appears (see below).

▶ To open a project method from the Explorer:

1 **Display the Methods page of the Explorer.**

2 **Expand the Project Methods item, then select the name of the method to be opened.**

3 **Double-click the name of the method to be opened.**
OR
Choose the Edit Method... command of the contextual menu — right-click on the name of the method to be opened.
The method opens in a window of the editor.

▶ To open a project method from the Menu Bar editor:

1 **In the Menu Bar editor, click in the "Method Name" area for an item of the current menu bar.**
This area contains the name of the method associated with the selected item (if any).

2 **Click on the edit button  to the right of the "Method Name" drop-down menu.**
The method is displayed in a new Method editor window.

The Menu Bar editor is described in the chapter "Custom Menus" on page 859.

▶ To open a project method from a window of the Method editor:

1 **In the Method editor, select the name of the project method to be opened.**

2 **Choose Edit Method in the contextual menu of the editor or type Ctrl+K (Windows) or Command+K (Mac OS).**

OR

1 **Alt+double-click (Windows) or Option+double-click (Mac OS) directly on the name of the project method (its name must not contain any spaces).**
The method is displayed in a new window of the Method editor. If 4D was not able to identify the method, the Methods page of the Explorer is displayed.

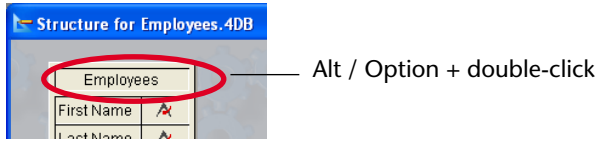
The Method editor is described in the section "Method Editor Window" on page 781.

Creating or Opening a Trigger

You can create a trigger using a shortcut in the Structure editor or you can create it directly in the Explorer:

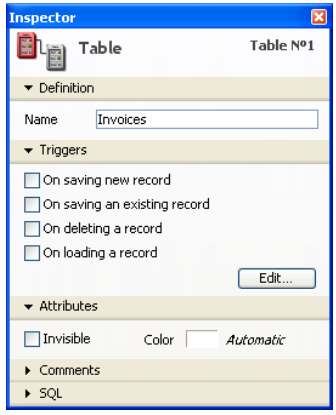
► To create or open a trigger from the Structure editor window:

1 **Hold down the Alt key (Windows) or Option key (Mac OS) and double-click the table title in the Structure editor window.**

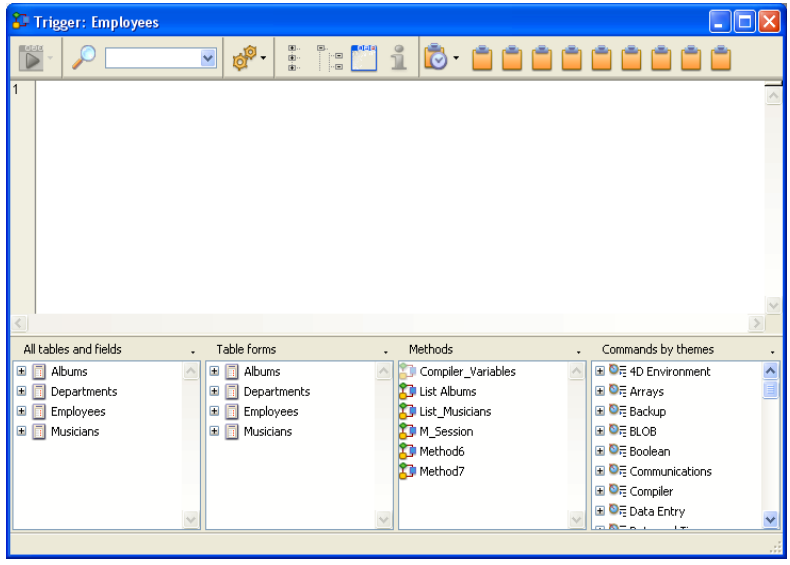


OR

In the Inspector window displaying the table properties, click on the Edit... button

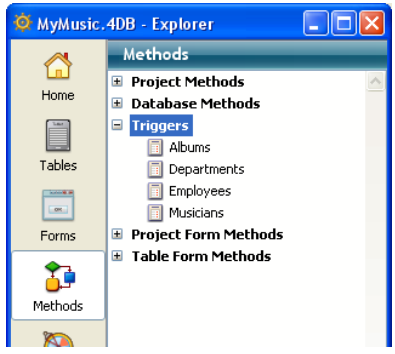


A Method editor window appears, blank if you have just created the trigger:



► To create or open a trigger from the Explorer:

1 Display the Methods page of the Explorer window and expand the "Triggers" item.



2 Select the table that interests you and click on the add button below the list.
OR
Double-click on the desired table.

The trigger appears in a window of the Method editor.

Creating or Opening a Form Method

Form methods can be opened from the Methods page of the Explorer or from the Form editor.

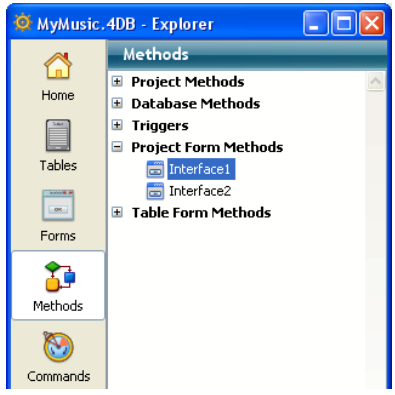
► To create or open a Form method from the Explorer:

1 **Display the Methods page of the Explorer.**

2 **Expand the "Project Form Methods" or "Table Form Methods" item, depending on the type of form for which you want to open the method.**

3 **(Table forms only): Expand the table to which the form belongs.**

4 **Select the form for which you want to open the method.**



5 **Click on the add button**  **below the list.**
OR
Double-click the form.
The form method appears in a window of the Method editor.

Note You can also create or open a form method from the **Forms** page of the Explorer: right-click on the name of the form and choose **Edit Form Method...** in the contextual menu.

► To create or open a form method from the Form editor:

1 **Choose Form Method from the Form menu.**
OR
Click an empty area on the form using the right mouse button and choose Form Method from the contextual menu.
OR
In the Property List, click the Edit... button located next to the Form Method line.
The form method appears in a window of the Method editor.

Deleting Methods

You can delete a trigger, form method or project method at any time using the Explorer.
You can also delete an object method using the Form editor.

On the other hand, you cannot delete database methods. To disable this type of method, erase all the statements in the method or change them to comments by preceding them with the "`" symbol (used to distinguish comments from executable code). For more information about comments in methods, refer to the "Comment/Uncomment" paragraph on page 806.

Deleting a Project Method, a Form Method or a Trigger

► To delete a form method, a project method, or a trigger using the Explorer:

1 **In the Explorer, display the Methods page.**

Make sure the Methods page is displayed, since certain deletion operations cannot be undone.

2 **In the Explorer window, expand the method type that corresponds to the method you want to delete (Project Form Methods, Table Form Methods, Triggers or Project Methods).**

3 **Select the table or form to which the method you want to delete is assigned.**
OR
Highlight the name of the project method you want to delete.

4D displays the method in the Method editor.

4 **Click the delete button located at the bottom of the Explorer window.**

Note To delete a project method, you can also **right-click** on the name of the project method and choose **Delete Method** from the contextual menu.

- If you delete a form method or a trigger, 4D displays a warning message asking you to confirm the operation. This deletion will be permanent.
- If you delete a project method, 4D carries out the operation directly. However, this deletion is not permanent so long as you have not emptied the Trash (for more information, refer to the "Trash Page" paragraph on page 129).

Clearing Unwanted Object Methods

You can clear unwanted object methods using the Form editor. In some cases, clearing unneeded object methods can make the database run faster.

► To delete an unwanted object method:

1 **Display the form that contains the method(s) you want to clear.**

2 **Select the object(s) that have unwanted object methods attached to them.**

3 **Choose Clear Object Method from the Object menu.**
4D removes the object methods from the selected objects.

Note If you clear an object method by mistake, choose **Undo** from the **Edit** menu.

Defining the Properties of Project Methods

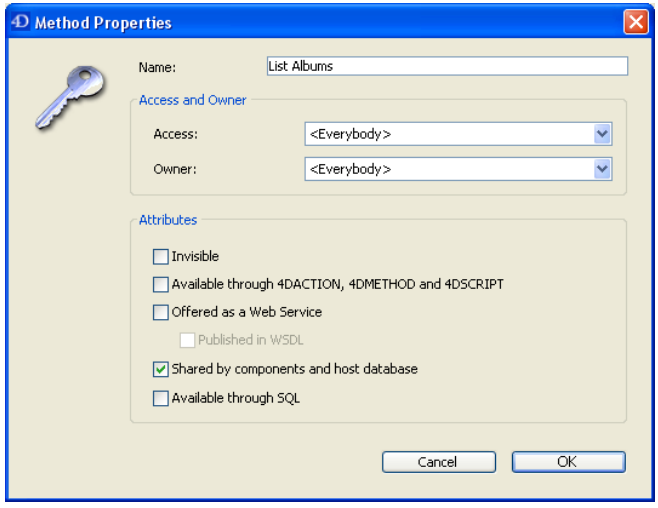
After creating a project method, you can rename it and modify its properties. Project method properties mainly concern their access and security conditions: user access as well as access by the Web server and Web services.

The other types of methods do not have specific properties. Their properties are related to those of the objects to which they are attached.

► To modify the properties of a project method:

1 **In the Method editor, select the Method Properties... command in the Method menu.**
OR
In the Method editor, right-click and choose the Method Properties... command from the contextual menu.
OR
On the Methods page of the Explorer, right-click on the project method and select Method Properties... in the contextual menu or options menu.

The Method Properties dialog box appears.



2 Modify the desired properties.

These properties are detailed in the following paragraphs.

Note A batch setting function can be used to modify a property for all or part of the database project methods in a single operation. This function is described in detail in the "Batch Setting for Method Attributes" paragraph on page 777.

3 Click OK to validate the modifications.

Renaming a Project Method

You can change the name of a project method in the Explorer. Database methods cannot be renamed. The same goes for triggers, form methods, and object methods, which are bound to objects and take their names from the object concerned.

► To rename a project method:

1 Display the Method Properties dialog box.

2 Modify the method name and click on OK.

You can also rename a method using the Explorer.

► To rename a project method using the Explorer:

1 Display the Methods page of the Explorer and expand the list of project methods.

2 **Hold down the Option key (Mac OS) or Alt key (Windows) and click the project method name.**
OR
Click twice (leave a gap between the two clicks) on the project method name.
The method name becomes editable.

3 **Type a new name.**

4 **Press Tab or click anywhere outside the entry area to save your changes.**
If a method with the same name already exists, 4D displays a message saying that the method name has already been used. Otherwise, 4D changes the name of the method and, if necessary, resorts the list of methods.

Note Changing a method name can be done in the Method Properties dialog box. This dialog is described in the following section. Changing a method name can invalidate any methods or formulas that use the old method name. Each such item has to be updated in order to work.

4D Server The method name is changed on the server when you finish editing it. If more than one user is modifying the method name at the same time, the final method name will be the name specified by the last user to finish editing it. You may want to specify a method owner so that only certain users can change the method's name.

Access and Owner Privileges

You can control access to methods by setting Access and Owner privileges for groups of users. A single group can be assigned for each privilege using the Access and Owner drop-down list in the Method Properties dialog box. For information about creating a password access system with users and groups, see the section "An Access Hierarchy Scheme" on page 836.

The Access drop-down list controls which group can execute the method. If a user that is not in this group attempts to execute the method, 4D displays a message saying that the user's password does not allow them to execute the method.

The Owner drop-down list controls which group can edit the method in the Design environment. If a user who is not in this group attempts to edit the method in the Design environment, 4D displays a message saying that the user does not have the access privilege to edit the method.

Users who are assigned to both groups can use or modify the method without restriction.

Making a Method Invisible

If you do not want users to be able to run a project method using the **Method...** command of the **Run** menu, you can make it Invisible in the Method Properties dialog box. An invisible method does not appear in the method execution dialog box (see the "From the Execute Method Dialog Box" paragraph on page 828).

When you make a project method invisible, it is still available to database programmers. It remains listed on the Methods page of the Explorer and in the list of routines in the Method editor.

Note 4D lets you modify this option for a large number of methods using the **Batch setting of attributes...** option. For more information, refer to the "Batch Setting for Method Attributes" paragraph on page 777.

Available through 4DACTION, 4DMETHOD and 4DSCRIPT

This option can be used to reinforce 4D Web server security: when it is not checked, the project method cannot be executed via an HTTP request containing the special URLs (4DACTION and 4DMETHOD) used for calling 4D methods, or the special 4DSCRIPT, 4DVAR and 4DHTMLVAR tags. For more information about this, refer to the *4D Language Reference* manual.

The project methods available are given a specific icon:  .

For security reasons, this option is unchecked by default. Methods that can be executed using the 4DACTION or 4DMETHOD Web URLs or the special 4DSCRIPT, 4DVAR and 4DHTMLVAR tags must be indicated individually.

Note 4D lets you modify this option for a large number of methods using the **Batch setting of attributes...** option. For more information, refer to the "Batch Setting for Method Attributes" paragraph on page 777.

Offered as a Web Service

This attribute lets you publish the current method as a Web Service via a SOAP request. For more information, refer to the "Publishing a Web Service with 4D" paragraph on page 1142.

When this option is checked, the **Published in WSDL** option is enabled.

Project methods that are offered as a Web Service are given a specific icon:  .

Published in WSDL

This attribute lets you include the current method in the WSDL of the 4D application. For more information about this, refer to the "Generation of the WSDL" paragraph on page 1145.

Project methods that are offered as a Web Service and published in WSDL are given a specific icon:  .

Note 4D lets you modify this option for a set of methods using the **Batch setting of attributes...** option. For more information, see the following paragraph.

Shared by Components and Host Database

This attribute is used within the framework of components. When it is checked, it indicates that the method will be available to components when the application is used as the host database. On the other hand, when the application is used as a component, the method will be available to the host databases.

For more information about components, please refer to the chapter "4D Components" on page 1269.

Available through SQL

When it is checked, this option allows the project method to be executed by the SQL engine of 4D. By default, it is not selected, which means that, unless explicitly authorized, 4D project methods are protected and cannot be called by the SQL engine of 4D.

This property applies to all internal and external SQL queries — executed via the ODBC driver, SQL code inserted between the Begin SQL/End SQL tags or the QUERY BY SQL command.

Notes
- Even if a method has the "Available through SQL" attribute, access rights set at the level of the Preferences and method properties are taken into account for the execution of the method.
- The ODBC *SQLProcedure* function only returns project methods with the "Available through SQL" attribute.

For more information about using the SQL engine of 4D, please refer to the *4D SQL Reference* manual.

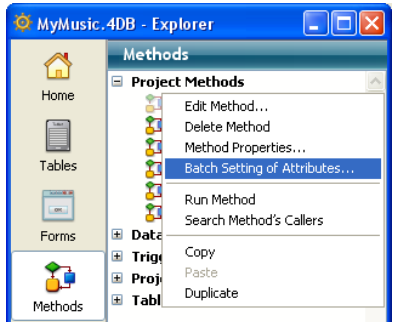
Batch Setting for Method Attributes

4D offers a function that can be used to modify an attribute (Invisible, Available through 4DACTION, 4DMETHOD and 4DSCRIPT, etc.) for all or part of the database project methods in a single operation.

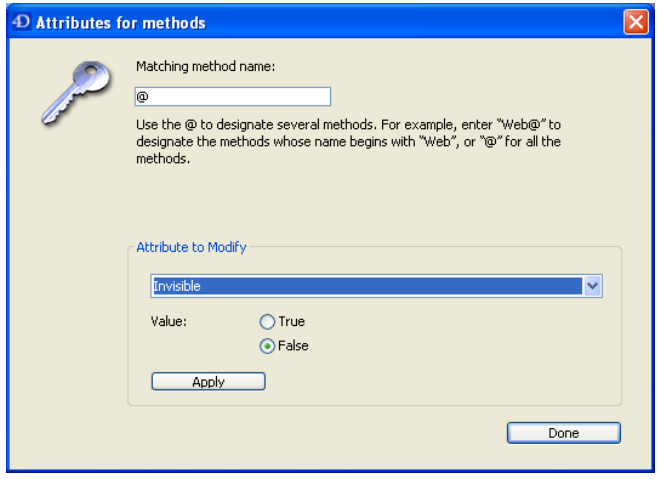
This new feature is especially useful for modifying the attributes of a large number of project methods. It can also be used during development to apply common attributes to groups of similar methods quickly.

► For batch setting of method attributes:

1 In the 4D Explorer, click with the right button, then choose the Batch setting of attributes... command.



The "Attributes for methods" dialog box appears:



2 In the "Matching method name:" area, enter a string that lets you designate the methods you want to modify as a batch.

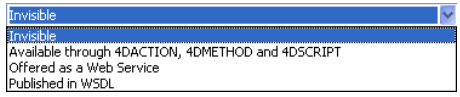
The character string is used as a search criterion for the method names. Use the wildcard character @ to help define groups of methods:

- To designate methods *whose names begin with...*, type @ at the end of the string. For example: web@.

- To designate methods *whose names contain...,* type @ in the middle of the string. For example: web@write.
- To designate methods *whose names end with...,* type @ at the beginning of the string. For example: @write.
- To designate all of the methods, just type @ in the area.

Notes • The search does not take upper/lower case into account.
• You can enter several @ characters in the string, for example dtro_@web@pro.@

3 In the "Attribute to Modify" area, choose an attribute from the drop-down list, then click on the True or False radio button corresponding to the value to be applied.

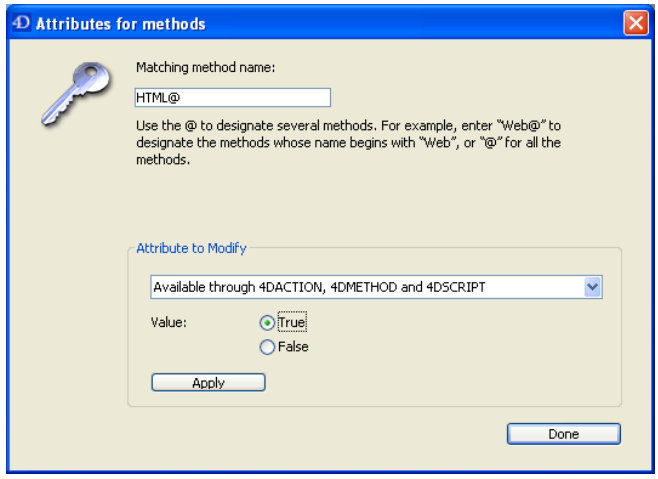


All of the different method attributes can be modified for a given batch:

- Invisible
- Available through 4DACTION, 4DMETHOD and 4DSCRIPT
- Offered as a Web Service
- Published in WSDL.

Note If the "Published in WSDL" attribute is set to True, it will only be applied to project methods already containing the "Offered as a Web Service" attribute.

In this example, the “Available through...” option will be checked for all the methods that have names beginning with “HTML”:



4 **Click on Apply.**

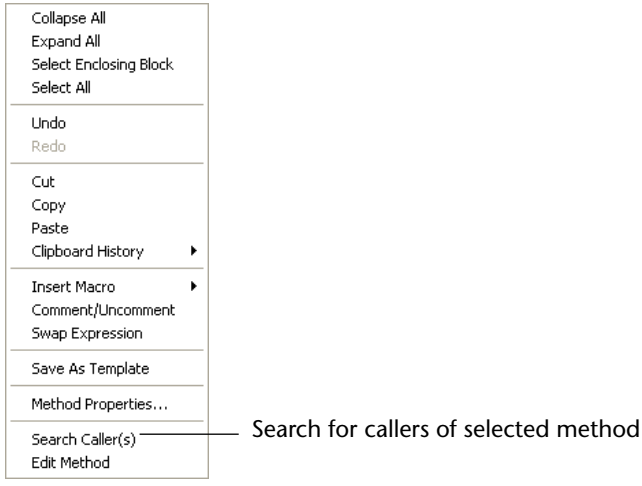
The modification is applied instantly to all the project methods designated by the character string entered.

5 **Repeat the operation for each batch of methods and/or each attribute to be applied.**

6 **Click Done to close the dialog box.**

Search Caller(s)

The contextual menu of the Method editor provides a **Search Caller(s)** function which is only enabled when a submethod is selected in the Method editor:

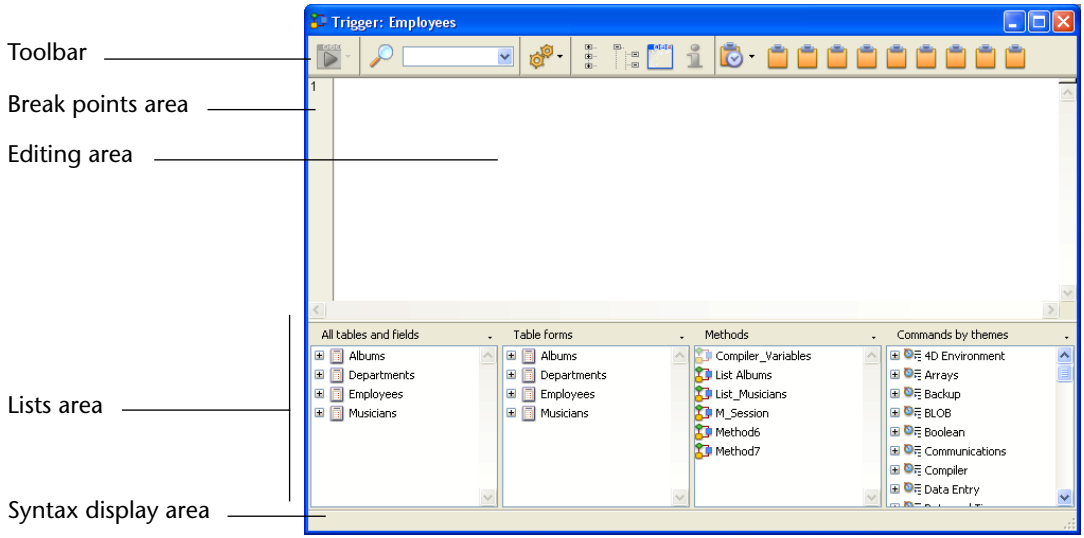


This function displays a search results window containing the list of objects (other methods or menus) that reference the selected sub-method (the objects found are displayed in the Callers list).

Note This function is also available in the 4D Explorer (see the "Searching for Callers" paragraph on page 122).

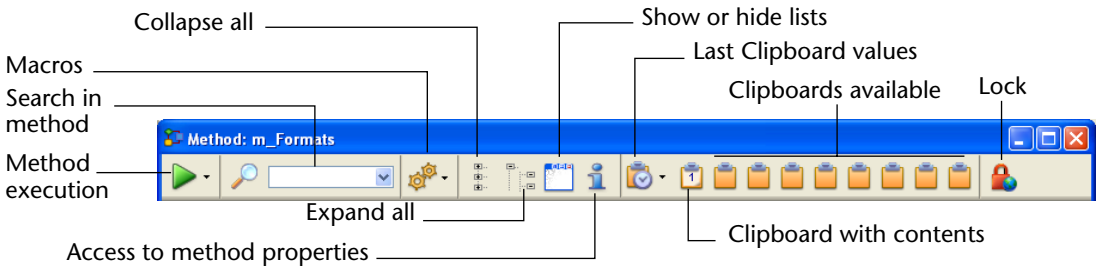
Method Editor Window

By default, a window of the Method editor consists of five areas: a toolbar, an editing area, a lists area, a break points area and a syntax display area.
The figure below shows the default appearance of a Method editor window.



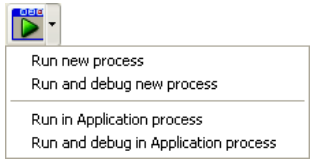
Toolbar

Each Method editor window contains a toolbar that provides instant access to basic functions related to method execution and editing:



The toolbar functions are as follows:

- **Method execution**: This button causes the execution of the method. Using the menu associated with the button, you can select the type of execution:



  Only project and database methods can be run using this button. For more information about these commands, please refer to the "Executing Methods" paragraph on page 827.

- **Search**: Clicking on the search icon causes the display of the standard search dialog box, which allows performing a search in the current method.
  The associated combo-box can be used to launch a standard search directly: to do this, enter the character string you want to search for and press **Enter** or the **Carriage return**. The combo-box also lists the last searches carried out; to redo a search, select it from the combo-box menu.
  For more information, refer to the "Find and Replace in Methods" paragraph on page 821.

- **Macros**: This button displays a menu listing all available macro-commands.
  For more information, refer to the "Creating and Using Macros" paragraph on page 813.

- **Expand all** / **Collapse all**: These buttons allow expanding or collapsing all the control flow structures of the method.
  For more information, refer to the "Expand/Collapse" paragraph on page 803.
- **Show or hide lists**: This button allows displaying or hiding the lists in the window.
  For more information, refer to the "Lists Area" paragraph on page 784.
- **Method information**: This button causes the display of the Method Properties dialog box (project methods only). For more information, refer to the "Defining the Properties of Project Methods" paragraph on page 773.
- **Last Clipboard values**: This button displays a menu listing the last 20 items copied in the window. If you select an item, it is recopied at the spot where the cursor is located.
- **Clipboard**: These 9 icons represent the 9 clipboards available in the Method editor. A white icon containing a number indicates that a clipboard contains data; an orange icon indicates an empty clipboard.
  - To copy text to a clipboard, select it in the method then use **Shift+click** on the icon of the clipboard to be filled or use the **Ctrl+Shift+clipboard number** (Windows) or **Control+Shift+clipboard number** (Mac OS) shortcuts. Remember that you must use the numbered keys of the alphanumeric keyboard.
  - To paste the contents of a clipboard where a cursor is located, click on its icon or use the **Ctrl+clipboard number** (Windows) or **Control+clipboard number** (Mac OS) shortcuts. Remember that you must use the numbered keys of the alphanumeric keyboard.

  For more information, refer to the "Multiple Copy-Paste and Numbering of Clipboards" paragraph on page 805.
- **Lock** (4D Server only): This icon indicates that the method is locked by another user. Clicking the padlock allows reloading the method (and thus displaying any changes made by the other user).

Editing Area

The editing area contains the text of the method. You enter and modify the method text in this area. The editor automatically indents method text for clear program structure. You can include comments inside the method text for reference.

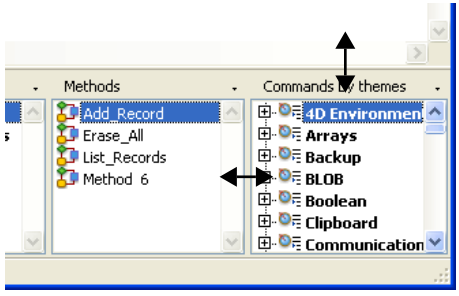
You can customize the display of this area. To do this, refer to the following paragraphs.

For more information about the entry of code in this area, refer to the "Writing a Method" paragraph on page 794.

Lists Area

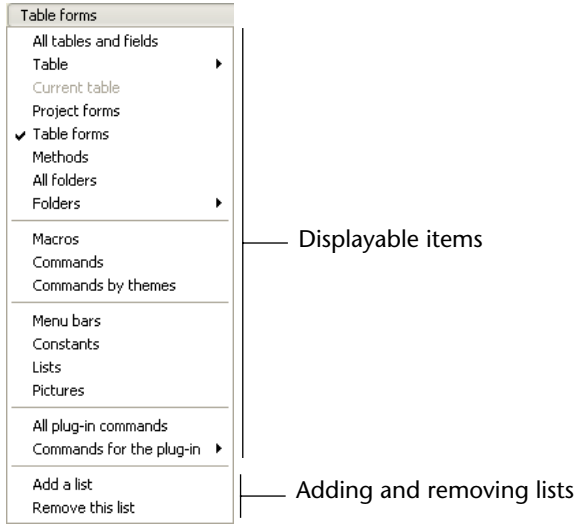
The lists area lets you display one or more lists of elements necessary for writing methods (commands, constants, forms, etc.). You can choose the number and contents of the lists displayed in the window.

By default, the Method editor displays four lists. You can enlarge or reduce the width of each list area by dragging one of its partitions. It is also possible to adjust the size of the list area in relation to that of the editing area by dragging the dividing line between them:



- **Double-clicking** on an item in a list causes it to be inserted into the editing area, at the location of the cursor.

- To **modify the contents** of a list, click on the title area of the list concerned: a pop-up menu appears, enabling you to choose the type of item to be displayed.

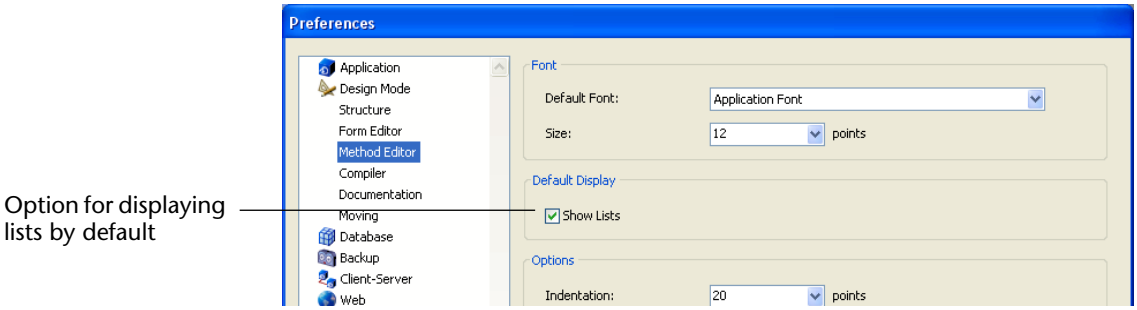


  For more information about the types of items that can be displayed, refer to the "Description of the Types of Lists" paragraph on page 787.

- For **adding** or **removing a list**, click in the title area of one of the lists and choose the corresponding command in the pop-up menu (see below).
  Note that at least one list must be displayed in the editor window. The **Remove this list** command is disabled when you click on the last list. If you want to hide all the lists, you must use the database Preferences dialog box (see above).

- You can display only the editing area of the Method editor window. To do this, click on the button used to show/hide the lists alternately in the toolbar of the window.

- You can also hide the lists by default. To do this, uncheck the **Show Lists** option on the **Method Editor** page of the database Preferences dialog box (“Design Mode” theme).



Any open methods must first be closed then reopened for the modifications made in the Preferences dialog box to be taken into account.

Note You can save the parameters set in the Method editor window in the form of a template. For more information, refer to the “Save As Template” paragraph on page 793.

Break Points Area

This area allows you to insert break points directly next to specific instructions. Break points are useful during the debugging phase of your programming. They stop the execution of your code at specific locations and display the debugger.

To insert a break point, click in the break points area at the location you want it to be placed. A red dot indicates the presence of a break point.

Note You can display the location of all the break points present in the database in the Runtime Explorer. For more information, refer to “Break and Catch Pages” paragraph on page 140.

To temporarily disable a break point or modify its properties, press the **Alt** key (Windows) or the **Option** key (Mac OS) while clicking on the break point. When you click the break point, the break point property window is displayed.

To delete a break point, click the red dot.

Note Break points are associated with line numbers. They remain at the same set location, even if you insert or delete a line.

For a complete description of break points, please refer to the *4D Language Reference* manual.

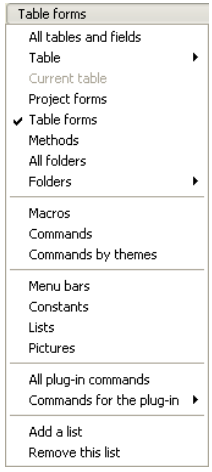
Syntax Display Area

This area displays the syntax of the 4D commands — the command name, followed by the parameter names — that you want to insert in your code. The syntax is displayed automatically when you enter a command manually. To "force" the display of the command syntax, insert a space after its name.

The syntax display area is also used to display potential syntax errors detected by 4D when the method is validated. For more information, refer to "Writing a Method" on page 794.

Description of the Types of Lists

You can display different lists of items in the lists area of the Method editor window.



This paragraph details each type:

- **All tables and fields**: Database table and field names in the form of a hierarchical list. When you insert a field name into the method by double-clicking on its name, 4D inserts it while respecting the syntax and adds the name of the table or subtable as the case may be.
- **Table** (submenu): Field names of the table selected using the submenu.
- **Current table**: Field names of the current table (available in triggers, form methods and object methods).
- **Project forms**: Database project form names. For more information about project forms, please refer to the "Table Forms and Project Forms" paragraph on page 339. When you double-click on a project form name, 4D inserts its while respecting the syntax: the form name is inserted between quotes.
- **Table forms**: Database table and form names in the form of a hierarchical list. When you insert a form name into a method by double-clicking its name, 4D inserts it while respecting the syntax: the form name is inserted between quotes and is preceded by the name of the table and a semi-colon. For example: [Table];"Form".
- **Methods**: Database project method names.
- **All folders**: Names of object folders and subfolders set in the database displayed in the form of a hierarchical list.
  Folders can be used to organize objects in a customized manner. They are managed from the Home page of the Explorer. For more information, refer to the "Home Page" paragraph on page 103.
- **Folders** (submenu): Contents of the folder selected using the submenu.
- **Macros**: Macro names defined for the database.
- **Commands**: 4D language commands in alphabetical order.
- **Commands by themes**: 4D language commands classified by theme in the form of a hierarchical list.
- **Menu bars**: Names and numbers of menu bars created with the 4D Menu bar editor (see the chapter "Custom Menus" on page 859).
- **Constants**: 4D constants and those of any plug-ins, classified by theme in the form of a hierarchical list.
- **Lists**: Names of lists.

- **Pictures**: Names and numbers of pictures stored in the 4D Picture Library.
- **All plug-in commands**: Commands for all the plug-ins installed in the database, classified by theme in the form of a hierarchical list.
- **Commands for the plug-in** (submenu): Commands of a specific plug-in selected using the submenu.
  By default, the following plug-ins are available:
  - **4D Internet Commands**: This plug-in adds additional Internet functions to 4D, in particular for the management of e-mail.
  - **4D Chart**: 4D Chart is a chart editor built into 4D.
  - **OLE_Tools** (Windows only): OLE_Tools, supplied with the Windows version of 4D, is a plug-in for the management of OLE areas.

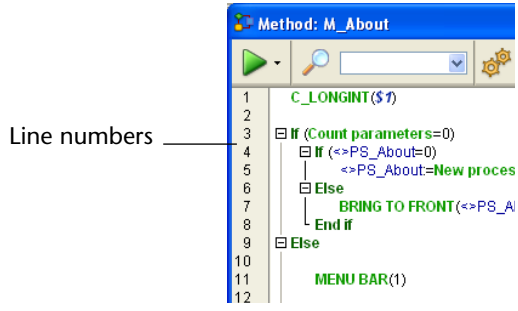
Notes
- These plug-ins have their own separate documentation.
- Except for the **Macros** element, all the lists are in alphabetical order.

Customizing the Window

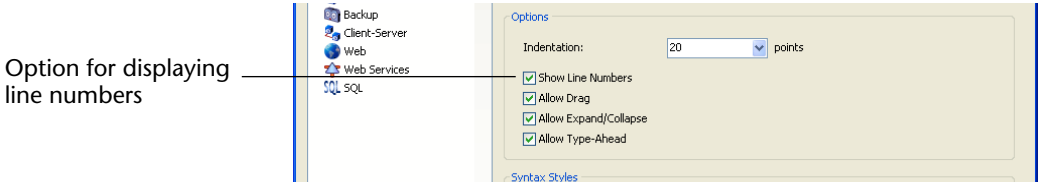
You can modify several display and operation options of the Method editor window.

Display of Line Numbers

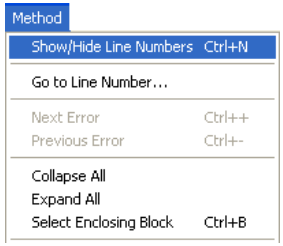
It is now possible to display the line numbers in each window of the Method editor:



The display of line numbers can be activated or deactivated by default using the **Show Line Numbers** option on the **Method Editor** page ("Design Mode" theme) of the database Preferences dialog box:



It is also possible to modify this display separately for each window of the Method editor, using the **Show/Hide Line Numbers** command in the **Method** menu:

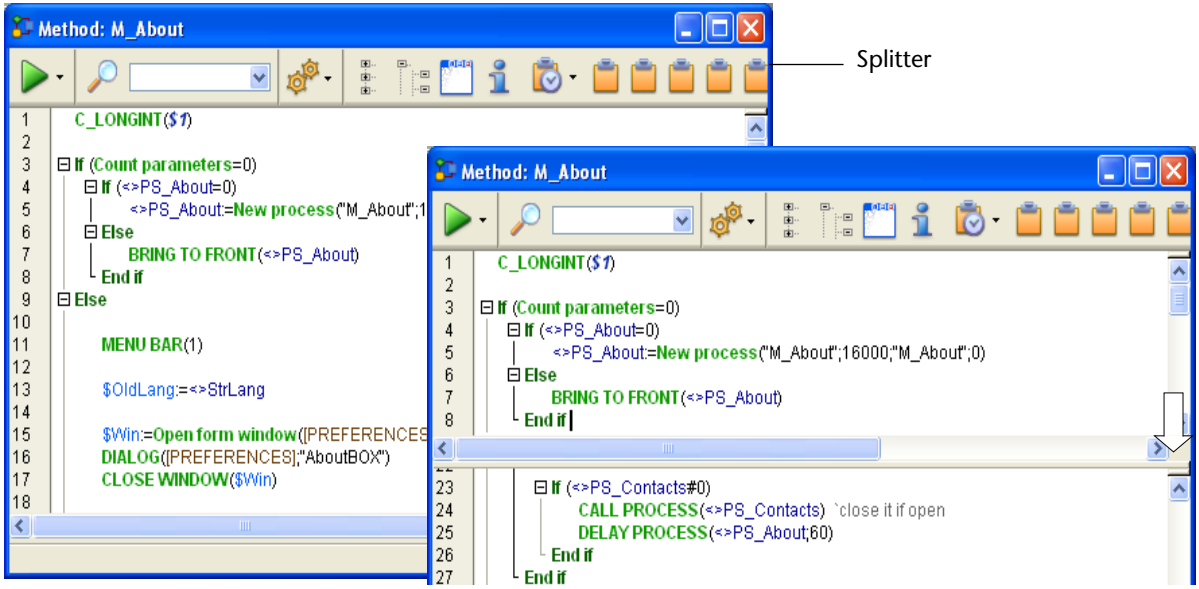


Displaying the line numbers makes it easier to find your way around in the window. The **Go to Line Number...** command in the **Method** menu also lets you take advantage of this display (see the "Go to Line Number" paragraph on page 824).

Splitting the Editor Window

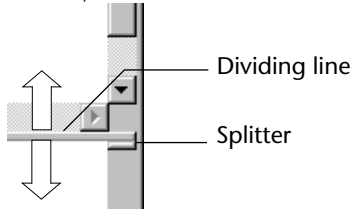
You can split the editing area into several horizontal panes. Once the editing area is split, you can view and scroll each part of the same method independently from the other parts. This is useful, for instance, when you want to keep the header of a method (generally containing the description of the method, as well as any comments or declarations of variables) on screen. It also enables simultaneous viewing of two or more distant areas of the same method.

- **Creating panes**
  To create a viewing pane, click on the splitter located at the top of the window and slide it towards the bottom:



To create several panes, simply repeat the operation as many times as necessary. You can create a pane below an existing pane by clicking (once) on the splitter and dragging it downwards.

- **Resizing a pane**
  To resize a pane, slide one of its dividing lines upwards or downwards. *Be careful* not to confuse the dividing line of a pane with its splitter:



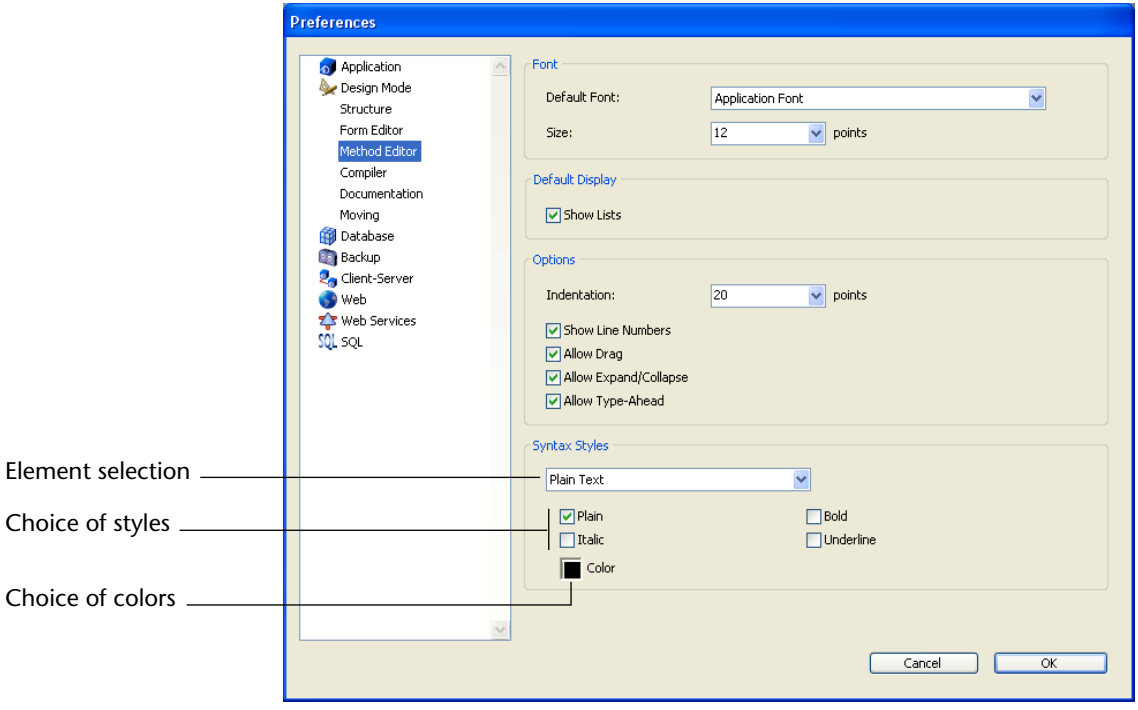
The splitter will not appear if the height of the pane is insufficient.

- **Removing a pane**
  To remove a viewing pane, slide its lower dividing line to the top of the window or double-click on this line.

Customizing Styles and Colors of Syntax Elements

It is possible to assign a specific color to each type of element of the 4D language (fields, tables, variables, parameters, etc.). The combination of different colors and styles for the method elements can prove to be extremely useful for code maintenance.

Styles and colors are set on the **Method Editor** page ("Design Mode" theme) of the application Preferences dialog box:



Notes

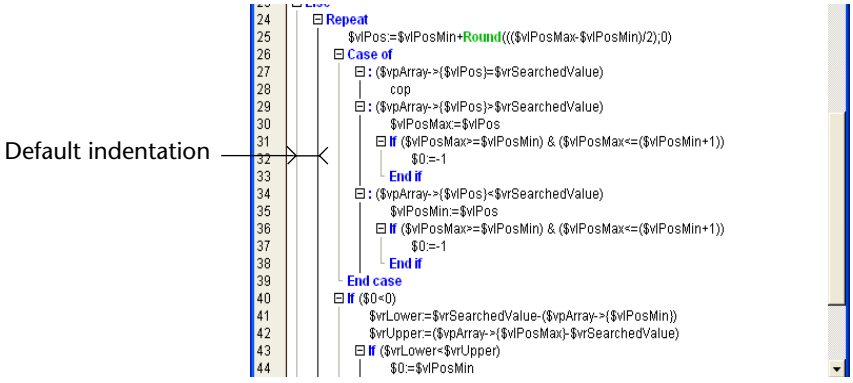
- The "Plain text" element type indicates all texts not belonging to any other defined type (i.e., symbols, punctuation, literal constants, etc.).
- The "Parameters" element type indicates the method parameters ($0; $1, etc.).

The following styles are available:

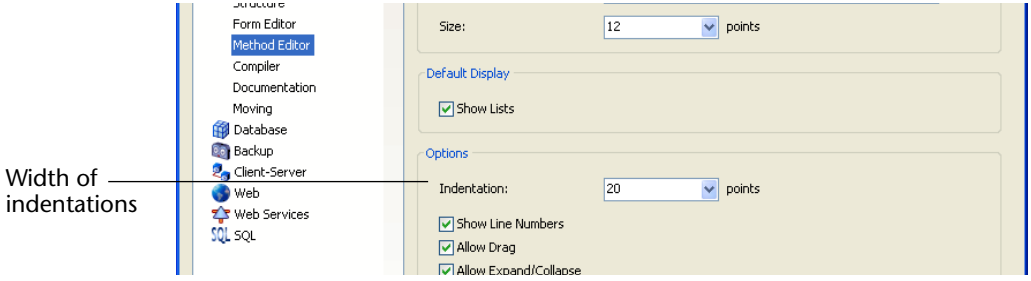
- **Normal**
- **Bold** (associated by default with 4D commands, keywords and plug-in commands).
- **Italic** (associated by default with plug-in commands and methods).
- **Underline** (associated by default with predefined constants).

Width of Indentations

The code is automatically indented in order to reveal its structure.



An option located on the **Method Editor** page (“Design Mode” theme) of the application Preferences dialog box enables you to modify the indentation width:



The width must be set in points (20 by default).
Modifying this default value can be useful if your methods contain complex algorithms with many levels of embedding. Narrower indentation can be used in order to limit horizontal scrolling.

Save As Template

You can save the parameters set in the Method editor window in the form of a “template.” Once the template is saved, the parameters set in it will be used for each new Method editor window that is opened. The following parameters are stored in the template:

- Relative size of the editing and list areas
- Number of lists
- Location and contents of each list
- Relative width of each list.

To save a Method editor window as a template, choose the **Save As Template** command in the **Method** menu or in the contextual menu of the Method editor:

Method editor contextual menu

The template is saved immediately (no dialog box appears). It is stored in the Preferences of the 4D application. If a previous template already exists, it is replaced.

Writing a Method

Writing a method is usually a combination of typing text, selecting components, and dragging items from the Explorer or other windows. You can also use various type-ahead functions to create methods faster.

The 4D Method editor provides basic syntax error-checking. Additional error-checking is performed when the method is executed.

You can customize the appearance and operation of the method editing area, for example by adding panes or displaying the line numbers.

Typing Text

4D uses standard text editing techniques for typing and editing in the Method editor. As you type, the characters appear at the location of the insertion point. You end each line by pressing the **Return** key (Mac OS) or **Enter**[1] key (Windows).

Note To enter a numeric value in hexadecimal, type 0x (zero + "x"), followed by the hexadecimal digits.

The Method editor uses default display conventions (style, color) for command names, plug-in commands, methods, etc. You can modify these conventions in the Preferences of the application. The font and font size can also be modified in the Preferences dialog box (refer to the "Customizing Styles and Colors of Syntax Elements" paragraph on page 792).

When you press the **Return**/**Enter** key, 4D evaluates the text of the line and applies the appropriate display format. 4D also indents each line to its proper level in relation to the preceding line when programming structures (for example If, End if) are used.

You can move the insertion point by clicking at the location you want. You can select words, whole lines, or several lines by dragging the I-beam over them.

You can use the arrow keys to move from line to line quickly. Using the arrow keys to move across several lines is quicker than clicking because the editor delays evaluating the line for errors.

Note The Method editor includes numerous navigation shortcuts. These shortcuts are listed in "Navigational Keyboard Shortcuts" on page 812.

Brace Matching

The Method editor has a brace matching option in the **Method** menu that helps you balance braces, parentheses, quotes, and brackets. A check mark appears next to the selected option.

1. The Enter key on the numeric keypad behaves differently from the Enter key on the main keyboard. Use the Enter key on the numeric keypad to force 4D to check the syntax of the line of code without moving the insertion point to the next line.

There are two levels of brace matching:

- **Small Brace Matching**: Affects only the opening and closing characters.
- **Big Brace Matching**: Affects the entire expression enclosed by the opening and closing characters.

When brace matching is active, 4D tries to find the matching brace, quote, or parenthesis when you type the closing character. When 4D finds matching characters, either the characters or the entire expression flashes.

For example, when you type

For ($i;1;**Records in selection** ([Line Items

and you press the closing bracket "]" to finish the table name expression, 4D will try to find the opening bracket. When it finds it, it flashes either the opening and closing brackets (Small brace matching) or the table name itself (Big brace matching).

As you continue to type the closing parentheses

For ($i;1;**Records in selection** ([Line Items]))

brace matching continues to provide feedback as you complete the arguments for the Records in selection function and the For keyword.

Turn off brace matching by choosing **No Brace Matching** from the **Method** menu.

Adding Method Objects by Drag and Drop

4D allows you to use the drag-and-drop mechanism when writing methods. It is possible to drag and drop items:

- From the Explorer,
- Within the same method,
- Between two methods.

Drag and Drop from the Explorer

From the Explorer, you can drag and drop:

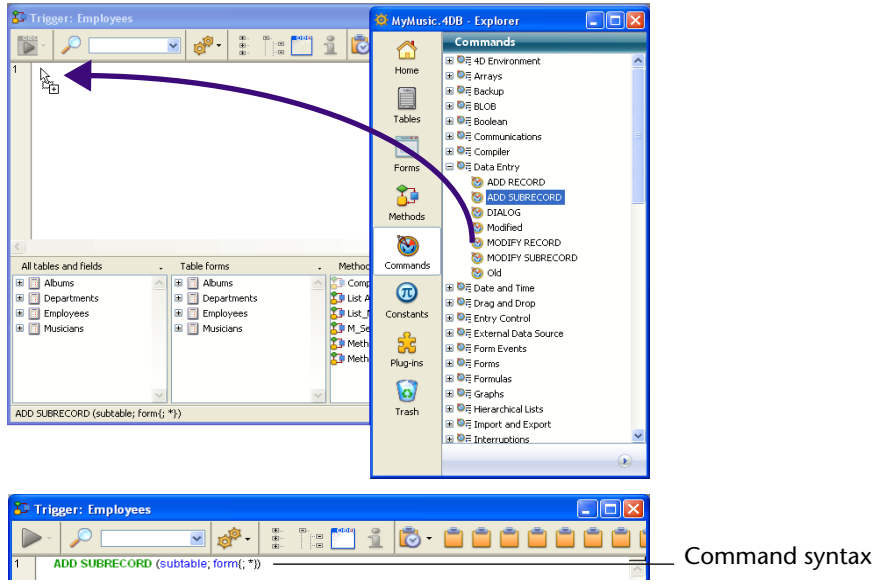
- Table names, field names, form names and project methods from the Home page,
- Table names and field names from the Tables page,
- Table names and form names from the Forms page,
- Project methods and form names from the Methods page,

- Constants from the Constants page,
- 4D commands from the Commands page,
- Plug-in commands from the Plug-ins page.

When you drag and drop a component, 4D always uses the correct syntax for the component. For example, if you drag the field name “First Name” from the [People] table, it appears in the Method editor as “[People]First Name.” Similarly, if you drag the Form name “Input” from the People table, it appears in the Method editor as “[People];"Input".”

When you insert a command by dragging it from the Explorer, it appears with its syntax (which consists of all of its parameters) in the Method editor.

Of course, you use the syntax that you need to adapt to your usage. This feature reminds you of the parameters that the command expects.



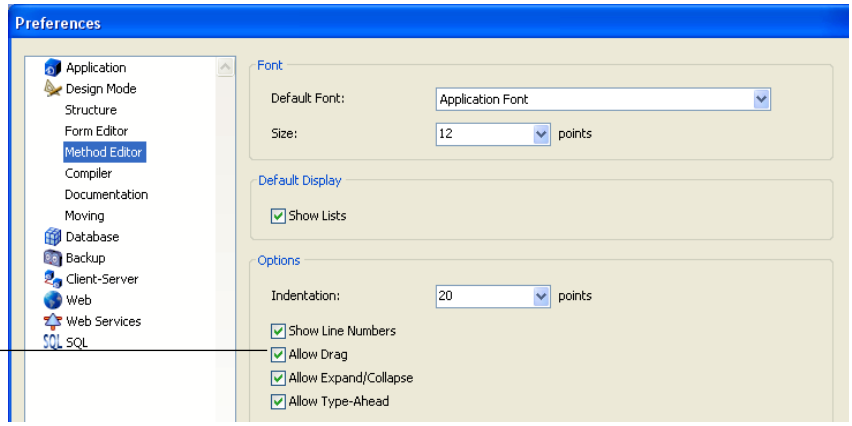
You can choose not to insert the syntax by pressing the **Alt** key (Windows) or the **Option** key (Mac OS) when dragging the command.

Drag and Drop Within a Method or Between Two Different Methods

In the Method editor, the drag-and-drop mechanism is activated as soon as a portion of text is selected. The cursor changes as shown here to indicate that the selection can be dragged and dropped:

- **Copying the selection**
  By default, the drag-and-drop mechanism moves the selected text. In order to copy it, hold down the **Ctrl** key (Windows) or the **Option** key (Mac OS) during the operation.

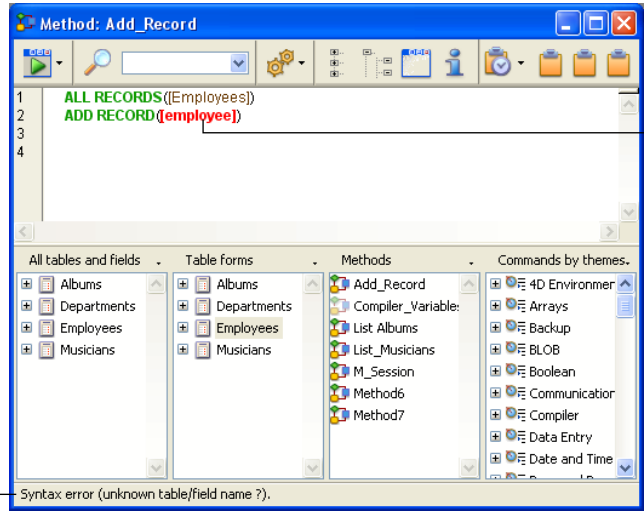
- **Disabling drag and drop**
  The drag-and-drop mechanism inside a method or between two methods can be enabled/disabled using the **Allow Drag** option (checked by default) located in the application Preferences dialog box, on the **Method Editor** page ("Design Mode" theme):



Note Inserting objects from the 4D Explorer using drag and drop remains possible even when the drag-and-drop mechanism is disabled inside the Method editor.

Checking and Correcting Syntax Errors

4D automatically checks the method syntax to see if it is correct. If you enter text or select a component that is not syntactically correct, 4D specifies the error in the syntax display area and indicates the incorrect expression, as shown in the following illustration.



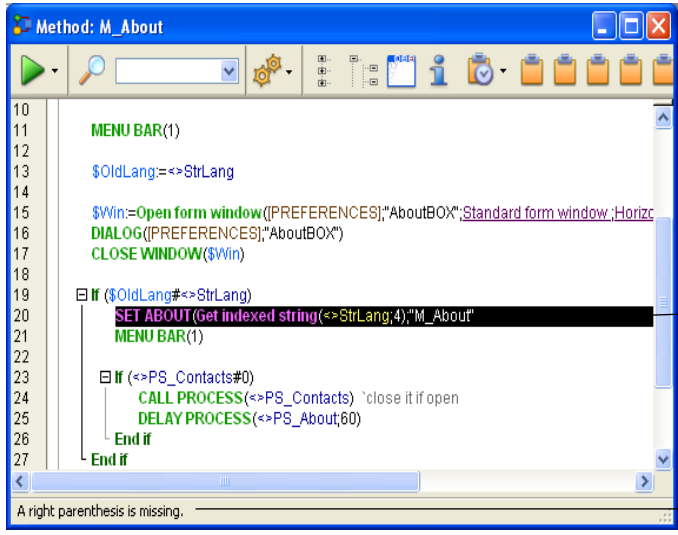
The style used to display the incorrect expressions can be modified in the Preferences of the application (see the "Customizing Styles and Colors of Syntax Elements" paragraph on page 792).

You can immediately check the syntax of the current line (without advancing to the next line) by pressing the **Enter** key on the numeric keypad. 4D evaluates the line, formats it, marks any errors, and places the insertion point at the end of the line.

When a line of a method is marked as having improper syntax, check and fix the entry. If the line is now correct, 4D returns to the standard style.

The validation of the entire method is done when you close the window. You can also force validation by pressing the **Enter** key.

When the method is validated, 4D checks for basic syntax errors and for the structure of the statements (**If**, **End if** and so on). When an error is detected, a message is displayed in the syntax display area and 4D highlights the line that contains the mistake.



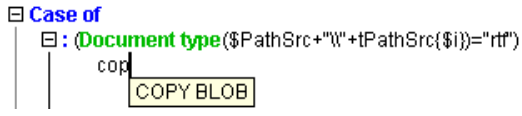
The Method editor can only check for obvious syntax errors (misspellings and the like). It does not check for errors that only occur during execution. Execution errors are caught by 4D when the method is executed. 4D provides a debugger for handling and correcting these errors. For information about the debugger, refer to the *4D Language Reference* manual.

The compiler also provides indispensable help for detecting errors. For more information about the compiler, refer to the chapter “Compilation” on page 1229.

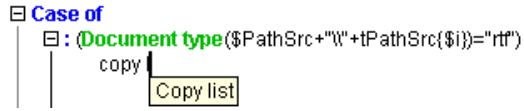
Type-ahead Function

The Method editor has a “type-ahead” function. 4D automatically displays propositions based on the first few characters typed.

In the example given below, typing the string “cop” causes the display of a tip containing the first 4D command (by alphabetical order) beginning with this string:

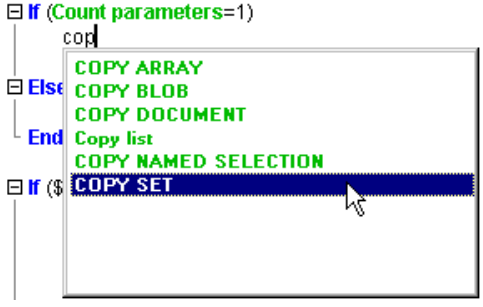


The value proposed in the tip is updated when additional characters are typed:



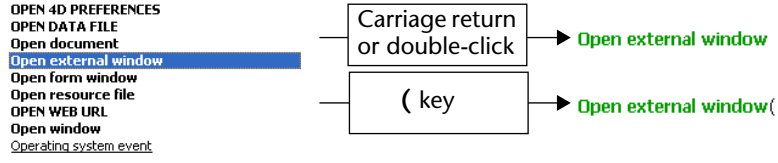
When the characters typed correspond to only one possible value, this value can be inserted by pressing the **Tab** key:

Otherwise, pressing the **Tab** key displays a list of all the words that begin with the characters typed:



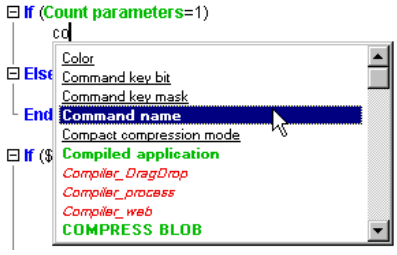
The list is in alphabetical order.

You can choose a value in the list by double-clicking on it or using the ⬆ and ⬇ arrow keys to scroll among the selections and then hitting the **Carriage return**.
You can also press the **(** (*open parenthesis*) or **;** (*semi-colon*) key after selecting a value: the value inserted is then followed by an open parenthesis or a semi-colon, ready for data entry.



You can also press the **Esc** key and continue typing.

If the characters typed correspond to different types of objects, the list displays them in their current style:



The following types of objects can be displayed:

- 4D commands
- User methods
- Table names
- Field names
- Constants
- Plug-in commands
- 4D keywords
- Macros.

Note The macro names are displayed between < >. For more information about macros, refer to the "Creating and Using Macros" paragraph on page 813.

Disable Type-Ahead

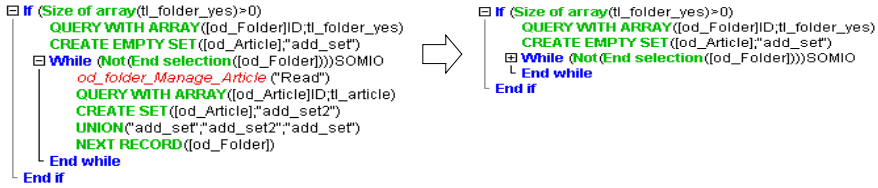
The type-ahead functions can be enabled/disabled using the **Allow Type-Ahead** option located in the application Preferences dialog box, on the **Method Editor** page ("Design Mode" theme).

Using the "Wildcard" (@)

Former versions of 4D used the wildcard character "@" as a shortcut when inserting object names. This mechanism can still be used, even though it is less powerful than the type-ahead mechanism. When you type several characters followed by the "@" character, 4D looks for element names beginning with these characters and completes the entry with the first name it finds that corresponds to the criteria. It is essential to enter enough characters to differentiate the desired word from similar ones since in the event of there being more than one possible solution, 4D may complete the entry by one of the other possibilities.

Expand/Collapse

4D code located inside loops and conditions can now be collapsed or expanded, in order to facilitate the reading of methods.



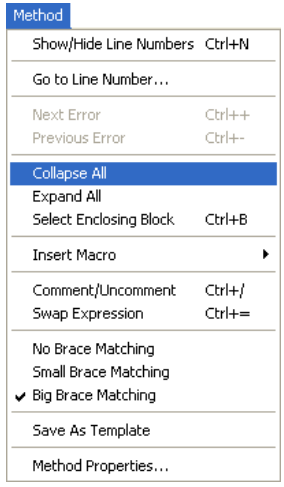
The operation of this new function is similar to that of hierarchical lists: clicking on the ⊟ icon causes the code included in the condition or loop to be collapsed. The ⊞ icon indicates a portion of collapsed code; clicking on this icon causes the portion of collapsed code to be expanded.

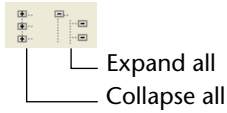
Note Conditions and loops (**If**, **Case of**, etc.) are defined with the help of macros. For more information, refer to the "Creating and Using Macros" paragraph on page 813.

When a portion of code is collapsed, it can no longer be modified. It is automatically expanded when you enter or delete a character in the condition or loop concerned.

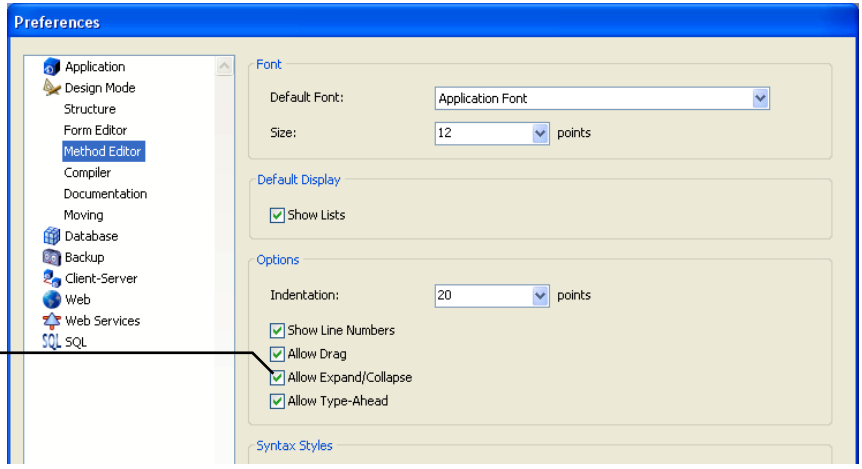
A collapsed portion of code can be selected, copied, pasted or deleted. All the lines included therein will be copied, pasted or deleted respectively. When a portion of code is pasted, it is automatically expanded.

It is possible to expand/collapse all of the loops and conditions of a method using the **Collapse All** and **Expand All** commands in the **Method** menu or in the Method editor contextual menu, or using the toolbar of the window.





The display of loops and conditions in the form of hierarchical lists can be activated or deactivated using the **Allow Expand/Collapse** option on the **Method Editor** page ("Design Mode" theme) of the database Preferences dialog box:



Activation of code hierarchical display option

This option is checked by default.

Note The display mode is not taken into account during printing of methods: as in previous versions of 4D, the source code is always expanded and vertical lines outline the conditions and loops.

Multiple Undo/Redo

The Method editor of 4D enables multiple undo/redo operations. All the actions performed (text entry, deletion, copy/paste, etc.) are stored sequentially in memory and can be undone in the reverse order of their execution. Likewise, each action undone can be redone.

4D thus records the last 20 actions carried out.

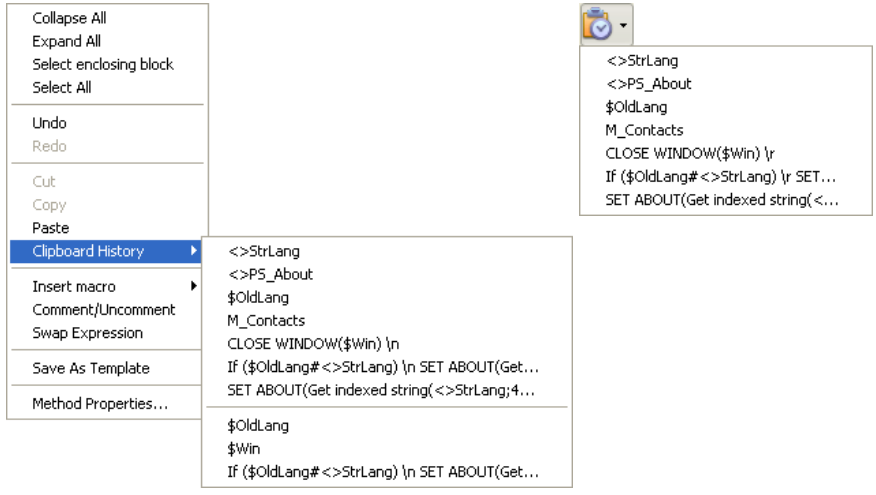
The **Undo/Redo** commands are available in the **Edit** menu and the Method editor contextual menu.

Multiple Copy-Paste and Numbering of Clipboards

In addition to the standard copy-paste operation, 4D proposes two additional functions that let you work with the contents of different clipboards:

- The program stores the last 10 “copy” or “cut” actions that were performed in the Method editor in memory during the current session. Each of the different contents saved in this way can be reused at any time.

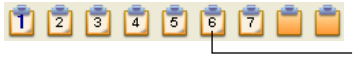
  To do this, use the **Clipboard History** command of the Method editor contextual menu or the “Last Clipboard values” button of the toolbar:



  The first few words of the copied or cut items are displayed. Selecting an item causes it to be inserted at the current location of the cursor.

- The 10 current clipboards are numbered and can be employed directly using a keyboard shortcut or the buttons of the Method editor toolbar:
  - To *copy* the selection into a specific clipboard, use **Ctrl+Shift+1** to **9** (Windows) or **Control+Shift+1** to **9** (Mac OS), or hold down **Shift** and click on one of the nine Clipboard icons:

  - To *paste* the contents of a specific clipboard, use **Ctrl+1** to **9** (Windows) or **Control+1** to **9** (Mac OS), or click on one of the nine Clipboard icons:



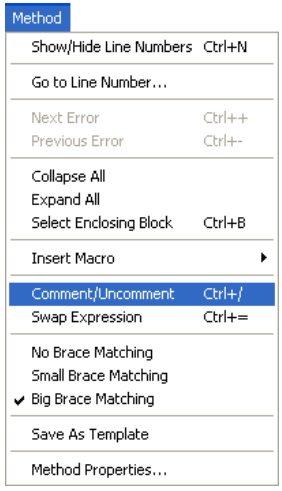
Comment/Uncomment

Comments are inactive lines of code. These lines are not interpreted by the program (4D does not apply any particular style within comments) and are not executed when the method is called.
To create a comment, you just need to insert the ` character at the beginning of a line.

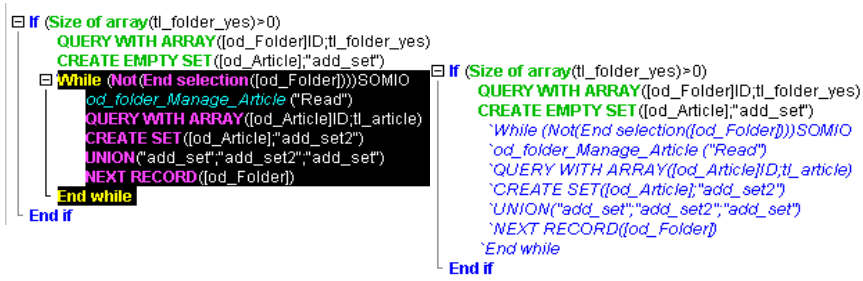
The length of comments is limited to the maximum size of a line, which is 32,000 characters.

The **Comment/Uncomment** command is used to mark a group of selected lines of code as comments, or, on the contrary, to remove the comment characters from a selection.

The **Comment/Uncomment** command is available in the **Method** menu as well as in the Method editor contextual menu.



To use this command, select the code to be marked as commented, then select the **Comment/Uncomment** command:



When the selection contains only active code, the **Comment** command is applied.

When the selection includes both active code and commented lines, an additional comment character (`) is added to the latter; this way, they will retain their initial commented status if the line is subsequently "uncommented."

When the selection contains only commented lines, the **Uncomment** command is applied.

Note The **Comment/Uncomment** command only operates with full lines — it cannot be used to comment only part of a line.

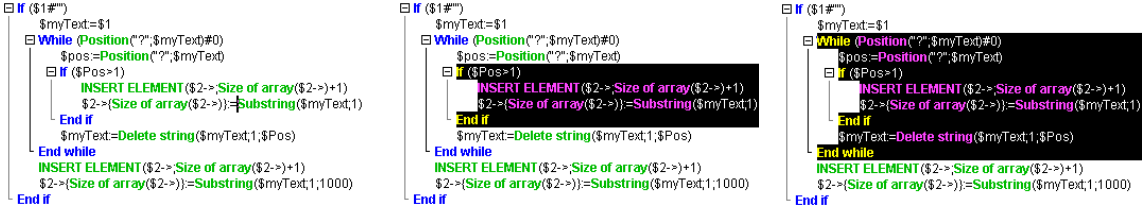
Selection of Enclosing Block

The **Select Enclosing Block** function is used to select the "enclosing block" of the code containing the insertion point. The enclosing block can be defined by:

- Quotes,
- Parentheses,
- A logical structure (If/Else/End if, While/End while, Repeat/Until Case of/End case), or,
- Braces.

If a block of text is already selected, the function selects the enclosing block of the next highest level and so on, until the entire method is selected.

In the following example, the **Select Enclosing Block** function is applied twice consecutively in order to select the desired block of code.



Pressing **Ctrl+Shift+B** (Windows) or **Command+Shift+B** (Mac OS) enables you to reverse this operation and deselect the last enclosing block selected.

Note If the insertion point is placed in an **If** or **Else** type structure, the enclosing block will be the one containing, respectively, the **If** or **Else** statement.

Swap Expression

The **Swap Expression** function can be used to reverse the arguments of an expression assigning a value. For instance,

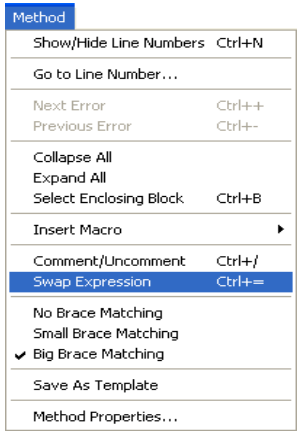
variable1:=variable2

becomes

variable2:=variable1.

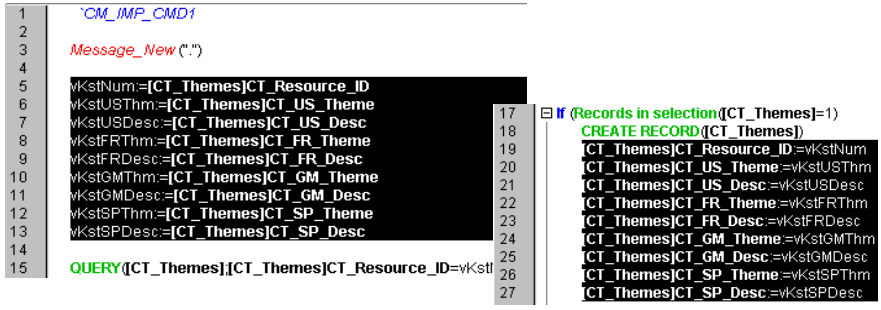
This function is extremely useful for reversing a set of assignments used to get or set properties, or to correct input errors.

To use this function, select the line(s) to be modified, then choose the **Swap Expression** command in the **Method** menu, or in the contextual menu of the area:



Within the selection, only the lines assigning a value will be modified.

In the import method shown below, the variable assignment area located at the beginning of the method is copied, then the expressions it contains are reversed using the **Swap Expression** command:



Managing of Long Strings

In the Method editor of 4D, character strings are limited to 80 characters. However, the entry of long strings is managed automatically: the editor divides strings that are too long and inserts the necessary syntax elements at the time of line validation:

Input:

```
1
2
3
ALERT("This character string is really much too long to be kept in its original format; it will be divided automatically by the method editor.")
```

After validation:

```
1
2
3
ALERT("This character string is really much too long to be kept in its original format; "+"it will be divided automatically by the method editor.")
```

Use of Escape Sequences

The Method editor allows you to use *escape sequences* (also called *escape characters*). An escape sequence is a sequence of characters that can be used to replace a “special” character.

The sequence consists of a backslash \, followed by a character. For instance, \t is an escape sequence for the Tab character. Escape sequences facilitate the entry of special characters: the previous example (\t) replaces the entry Character(Tab).

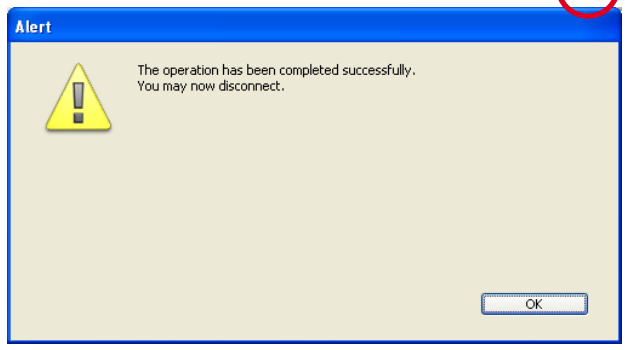
In 4D, the following escape sequences can be used:

Escape sequence	Character replaced
\n	LF (Line feed)
\t	HT (Tab)
\r	CR (Carriage return)
\\	\ (Backslash)
\"	" (Quotes)

Note It is possible to use either upper or lower case in escape sequences.

In the following example, the *Carriage return* character (escape sequence \r) is inserted in a statement in order to obtain the dialog box shown below:

```
ALERT("The operation has been completed successfully.\rYou may now disconnect.")
```



The same statement could have been written as follows:

```
ALERT("The operation has been completed successfully."+Char(Carriage return)+"You may now disconnect.")
```

Warning: The \ (backslash) character is used as a separator in pathnames under Windows. In general, 4D will correctly interpret Windows pathnames entered in the Method editor by replacing the single backslash \ with a double backslash \\. For instance, C:\Folder will become C:\\Folder.
However, if you write “C:\MyDocuments\New”, 4D will display “C:\\MyDocuments\New”. In this case, the second backslash is interpreted incorrectly as \N (an existing escape sequence). You must therefore enter a double backslash \\ when you want to have a backslash in front of a character used in one of the escape sequences recognized by 4D.

Navigational Keyboard Shortcuts

Helpful keyboard shortcuts to navigate the code are available in 4D's Method editor[1].

Windows	Mac OS	Action
[Shift]+[→]		Create and enlarge the selection, character by character, to the right, or Reduce the selection, character by character, from the left
[Shift]+[←]		Reduce the selection, character by character, from the right or Create and enlarge the selection, character by character, to the left
[Shift]+[↓]		Create and enlarge a selection, line by line, from the top to the bottom
[Shift]+[↑]		Create and enlarge a selection, line by line, from the bottom to the top
[Ctrl]+[Shift] +[→]	[Command]+ [Shift]+[→]	Create and enlarge the selection, word by word, from the right
[Ctrl]+[Shift] +[←]	[Command]+ [Shift] +[←]	Reduce the selection, word for word, from the right, or create and enlarge the selection, word by word, from the left
[Ctrl]+[→]	[Command]+[→]	Move the insertion point, word by word, from left to right
[Ctrl]+[←]	[Command]+[←]	Move the insertion point, word by word, from right to left
[Home]		Place the insertion point at the beginning of the line
[End]		Place the insertion point at the end of the line
[Ctrl]+[Home]	[Command]+ [Home]	Place the insertion point at the beginning of the method
[Ctrl]+[End]	[Command]+ [End]	Place the insertion point at the end of the method
[Shift]+[Home]		Select all the characters in the line that are to the left of the cursor
[Shift]+[End]		Select all the characters in the line that are to the right of the cursor

1. These shortcuts are also available in all of 4D's dialog boxes that contain data entry areas.

Windows	Mac OS	Action
[PgUp]		Scroll the contents of the method, page by page, from the bottom to the top (doesn't modify the insertion point)
[PgDn]		Scroll the contents of the method, page by page, from the top to the bottom (doesn't modify the insertion point)

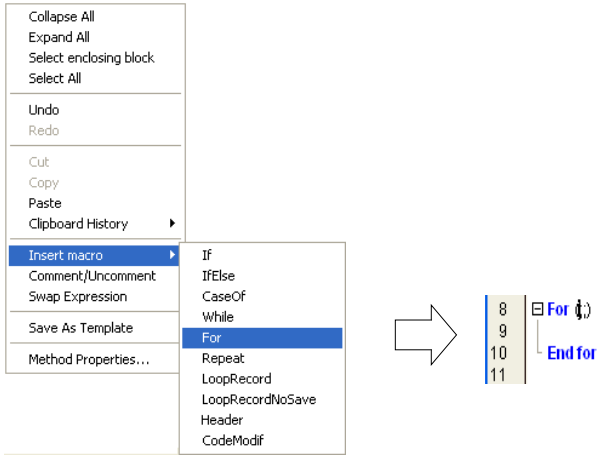
Creating and Using Macros

You can use macro-commands in your methods. Using macro-commands saves a lot of time during method entry.

What is a Macro?

A macro-command is a section of 4D code that is permanently accessible and that can be inserted anywhere in your methods, whatever the type of database open. Macros can contain all types of 4D text, commands and constants, as well as special tags which are replaced at the time of macro insertion by values derived from the method context. For instance, a macro may contain the tag <method_name/>; at the time of macro insertion, this tag will be replaced by the name of the current project method.

Example of macro insertion



Macros are stored in one or more XML format (text) file(s). They can be placed in a Method editor list; they can also be called using the contextual menu of the editor or using the type-ahead function.

4D macros are written in XML format. You can use the 4D default macro file "as is" or modify it.

Location of Macros

4D loads the macros from a folder named "**Macros v2**." Macros must be in the form of one or more XML files that are placed in this folder.

The "Macros v2" folder can be located:

- In the **active 4D folder** of the machine. Macros are then shared for all the databases.

Note The location of the active 4D folder varies according to the operating system used. For more information, refer to the description of the Get 4D folder command in the *4D Language Reference* manual.

- Next to the database structure file. Macros are only loaded for this structure.
- For components: in the Components folder of the database. Macros are then only loaded if the component is installed.

These three locations can be used simultaneously: it is possible to install a "Macros v2" folder in each location. The macros will be loaded in the following order: 4D folder, structure file, component 1... component X.

Default Macros

4D offers a set of default macros corresponding, in particular, to the list of keywords in previous versions of 4D. These macros are included in the default "Macros.xml" file, placed in the "Macros v2" folder that is created in the active 4D folder of the machine during the initial startup of 4D.

Note 4D allows you to embed programming structures (If/While/For/Case of) up to a "depth" of 512 levels.

You can modify this file or the contents of the folder subsequently as desired (see the following paragraph). In the event of problems with this folder, it can be deleted and 4D will re-create it on the next startup.

Adding Customized Macros

You can add customized macros in the “Macros.xml” file using a standard text editor or by programming. You can also add XML files of customized macros in this folder.

The macros file can be open while using 4D. The list of available macros is updated on each event activating 4D. For instance, it is possible to bring the text editor to the foreground, modify the macro file, then return to the method: the new macro is then available in the Method editor.

Empty or erroneous macros are not displayed.

Checking the Syntax of Customized Macros

The macro-command files of 4D must be in conformity with the XML standard. This means more particularly that XML declaration <?xml version="1.0" ...?> and document declaration < !DOCTYPE macros SYSTEM "http://www.4d.com/dtd/v11/macros.dtd"> statements are mandatory at the beginning of a macro file in order for it to be loaded. The different types of XML encoding are supported. However, it is recommended to use encoding that is Mac/PC (UTF-8) compatible. 4D provides a DTD that can be used to validate the macro files. This file is found in the following location:

- Windows: 4D Developer/Resources/DTD/macros.dtd
- Mac OS: 4D Developer.app:Contents:Resources:DTD:macros.dtd

If a macros file does not contain the declaration statements or cannot be validated, it is not loaded.

Incompatibilities Related to the XML Standard

Strict syntax rules must be observed in order for macros files to respect the XML standard. This may lead to incompatibilities with the code of existing macros and prevent the loading of XML files. The following are the main sources of malfunctioning:

- Comments of the “// my comment” type, allowed inside <macro> elements in previous versions of 4D, are not compatible with the XML syntax. The lines of comments must respect the standard “<!-- my comment -->” form.

- The <> symbols used more particularly for interprocess object names must be encoded. For example, the *<>params* variable must be written *<>params*.
- The initial <macros> declaration tag could be omitted in previous versions of 4D. It is now mandatory; otherwise, the file will not be loaded.

Syntax of 4D Macros

4D macros are built using customized XML tags called "elements."

Some tags indicate the start and end of the definition (double tags of the type <tag> </tag>), others are replaced by insertion context values (<tag/>).

In conformity with XML specifications, some element tags can include **attributes**. Unless otherwise indicated, these attributes are optional and a default value is used when they are omitted. The syntax of elements with attributes is as follows:

- Double tags: <tag attribute="value"> </macro>
- Single tags: <tag attribute="value"/>

If the element accepts several attributes, you can group them in the same line of command, separated by a space:
<tag attribute1="value" attribute2="value" attribute3="value"... >

Here is the list of tags and their mode of use:

Element tags	Description
<**macros**> </**macros**>	Start and end of macro file (mandatory tag).

Element tags	Description
<macro> </macro>	Start and end of the definition of a macro and its attributes. ***Attributes***: • **name**: Name** of macro as it appears in menus and Method editor lists (mandatory attribute). • **type_ahead_text**: Character string** to be entered to call the macro using the type-ahead function*. • **in_menu**: Boolean indicating whether the macro can be called using the contextual menu*. Values = "true" (default) or "false." • **type_ahead**: Boolean indicating whether the macro can be called using the type-ahead function*. Values = "true" (default) or "false." • **method_event**: Used to trigger the automatic calling of the macro depending on the current handling phase of each method (creation, closing, and so on). Values = "on_load": The macro is triggered on the opening of each method, "on_save": The macro is triggered when each method is saved (closing of a modified method or saving using the **File>Save** command, "on_create": The macro is triggered when each method is created, "on_close": The macro is triggered when each method is closed. "on_save" and "on_close" can be used jointly — in other words, both of these events are generated when a modified method is closed. On the other hand, "on_create" and "on_load" are never generated in a consecutive manner. This attribute can be used, for example, to preformat methods when they are created (comments in header area) or to record information such as the date and time when they are closed. • **version**: Used to activate the new mode of supporting text selections for the macro (see the "About the <method> Tag" paragraph on page 818). To activate this new mode, pass the value "2". If you omit this attribute or pass version="1", the former mode is kept.
<selection/>	Tag replaced by the selected text when the macro is inserted. The selection may be empty.
<text> </text>	Start and end of code that must be inserted in the method. A carriage return will be added before and after the code.
<method> </method>	Start and end of the name of the project method and its (optional) parameter. The method is executed when the macro is called. You can pass a parameter in the form ("param1;param2;..."). This parameter will be received in the method using the variables $1, $2, etc. For additional information about this tag, refer to the "About the <method> Tag" paragraph on page 818.
<caret/>	Location of the insertion point in the code after the macro has been inserted.
<user_4D/>	Tag replaced by the name of the current 4D user.
<user_os/>	Tag replaced by the current system user name.
<method_name/>	Tag replaced by the current project method name.

Element tags	Description
<**date**/>	Tag replaced by the current date. ***Attribute***: • **format**: 4D format used to display the date. If no format is set, the default format is used. Values = number of 4D format (0 to 8).
<**time**/>	Tag replaced by the current time. ***Attribute***: • **format**: 4D format used to display the time. If no format is set, the default format is used. Values = number of 4D format (0 to 6).
<**clipboard**/>	Tag replaced by the contents of the clipboard. ***Attribute***: • **index**: Clipboard to be pasted. Values = number of the clipboard (0 to 9).

***** Macros can be called using the contextual menu of the Method editor or using the type-ahead function (see the following paragraph).

****** If you want to conform to XML language specifications, you must not use extended characters (accented characters, quotes, etc.).

Here is an example of a macro definition:

Content of macro	Comments
<macros>	*Start of macros XML file*
<macro name="RecordLoop">	*Start of macro definition and name*
<text>	*Start of macro code*

For($i;1;Records in selection(<Selection/>))
SAVE RECORD(<Selection/>)
NEXT RECORD(<Selection/>)
End for
	The <Selection/> tag will be replaced by the selected code in the 4D method at the time of macro insertion (for instance, a table name)

</text>	*End of macro code*
</macro>	*End of macro definition*
</macros>	*End of macros XML file*

About the <method> Tag

The <method> tag allows you to generate and use macro-commands that execute 4D project methods. This allows developers to create sophisticated functions that can be distributed via macro-commands which are associated with components.

For example, the following macro will cause the *MyMethod* method to be executed with the name of the current method as parameter: <method>MyMethod("<method_name/>")</method>.

The code of a called method is executed in a new process. This process is killed once the method is executed.

The structure process remains frozen until the called method execution is completed. You must make sure that the execution is quick and that there is no risk of it blocking the application. If this occurs, use the Ctrl+F8 (Windows) or Command+F8 (Mac OS) command to "kill" the process.

Text Selection Variables for Methods (Compatibility Note)

In previous versions of 4D, the program automatically maintained a set of process variables for manipulating text in methods when using the <method> tag: input variables (*_textSel*, *_blobSe*l, *_selLen*, *_textMethod*, *_blobMethod*, *_methodLen*) to retrieve text and output variables (*_textReplace*, *_blobReplace*, *_action*) to insert text. For the sake of compatibility, this mechanism is still supported in 4D v11, but it is now obsolete for the following reasons:

- The use of BLOB variables for managing text with a size greater than 32,000 characters is no longer necessary,
- The management of variables is not compatible with the new component architecture in which the execution spaces of variables are partitioned. A version 11 component cannot access the text of host database methods (and vice versa) using predefined variables.

Consequently, it is now recommended to use the new mode for supporting text selections via the GET MACRO PARAMETER and SET MACRO PARAMETER commands. These commands can be used to overcome the partitioning of the host database/component execution spaces and thus allow the creation of components dedicated to the management of macros. In order to activate this new mode for a macro, you must declare the new **Version** attribute with the value **2** in the Macro element. In this case, 4D no longer manages the predefined variables *_textSel*, *_textReplace,* etc. and the GET MACRO PARAMETER and SET MACRO PARAMETER are used. This attribute must be declared as follows:

```
<macro name="MyMacro" version="2">
--- Text of the macro ---
</macro>
```

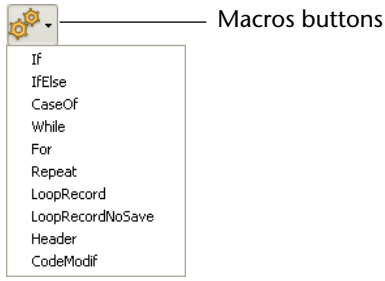
If you do not pass this attribute, the previous mode is kept.

Calling Macros

By default, macros can be called using the contextual menu or toolbar of the Method editor, the type-ahead function, or a specific list at the bottom of the Method editor window.
Note that for each macro it is possible to restrict the possibility of calling it using the contextual menu and/or the type-ahead function.

Contextual Menu and Toolbar

By default, all macros can be called via the contextual menu of the Method editor (using the **Insert macro** hierarchical command) or the "Macros" button of the toolbar.



The in_menu attribute of the <macro> tag is used to set whether or not the macro will appear in this menu.

In the contextual menu, macros are displayed in the order of the "Macros.xml" file and any additional XML files. It is thus possible to change the order by modifying these files.

Type-ahead

By default, all macros are accessible using the type-ahead function (see the "Type-ahead Function" paragraph on page 800). The macro replaces the entered text.
The type_ahead attribute of the <macro> tag can be used to exclude a macro from this type of operation.

Note If the macro contains the <selection/> tag, it will not appear in the type-ahead pop-up window.

Method Editor List

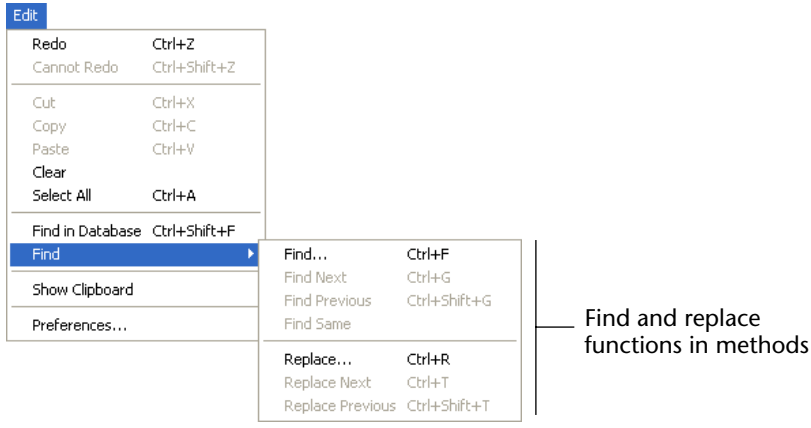
You can display your macros in a list of the Method editor (see the "Lists Area" paragraph on page 784). Simply double-click on the name of a macro in the list in order to call it.
It is not possible to exclude a specific macro from this list.

Find and Replace in Methods

The Method editor has specific find and replace functions that apply to the current window.

The find area located in the toolbar of each method window can be used to carry out simple searches or to call the Find dialog box (see the "Toolbar" paragraph on page 782).

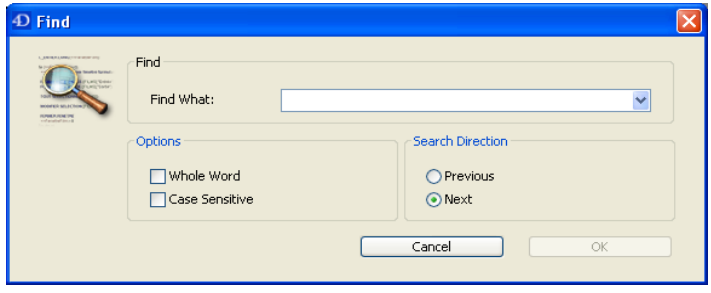
The Find/Replace commands for methods are located in the **Find** submenu of the **Edit** menu of 4D:



Note The **Find in Design** command enables an overall search in the database. It is not specific to the Method editor but may be used to search for a value among all the methods. For more information about this command, refer to the "Searching and Replacing in the Design" paragraph on page 140.

Find

Selecting the **Find...** command displays the following dialog box:



The search defined in this dialog box will be performed in the method located in the foreground.

- The "Find What:" area enables you to enter the string of characters to be searched for. This area is a combo box that stores the last 15 character strings that have been searched for or replaced during the session. If you highlight text before choosing the **Find...** command, it will appear in this area. You can then either use this text or replace it with another.
- The **Whole Word** option is used to limit the search to exact occurrences of the word being searched for. When this option is checked, for instance, a search for "client" will not find either "clients" or "mycli-ent." By default, this option is not checked; therefore, a search for "var" will find "Myvar," "variation," etc.
  Be careful, unlike the **Whole Object Name** option of the Find in Design dialog box, the **Whole Word** option does not take *object names* into account. For example, with this option, searching for the string "My" in a method will find the "My Variable" variable. This is not the case for an overall search using the **Whole Object Name** option, where the same result will not be found in the context of the above example since the whole object name (of the variable found previously) is "My Variable" and therefore does not correspond exactly to the string entered ("My").
- The **Case Sensitive** option is used to take the case of characters as they were entered in the "Find What:" area into account. For instance, a search for "MyVar" will not find "myVar."
- The **Previous/Next** radio buttons are used to set the direction of the search: towards the beginning or end of the current method, starting from the initial location of the cursor.

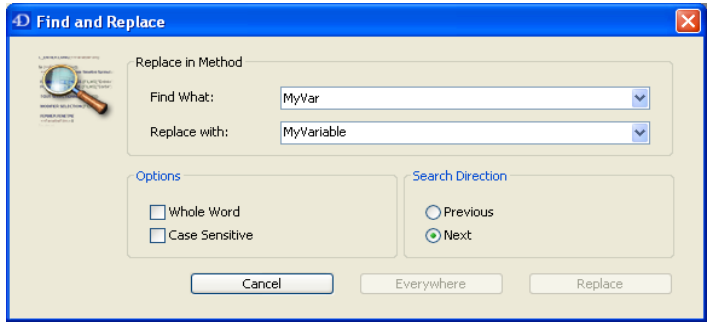
  The **OK** button is used to launch a search. 4D begins searching from the current text insertion point and continues to the end of the method. The first item corresponding to the set criteria is thus selected in the Method editor window. It is then possible to continue the search using the **Find Next** and **Find Previous** commands of the **Edit** menu.

Find Same

The **Find Same** command is used to find character strings identical to the one selected. This command is only active if you have selected at least one character in the Method editor.
The search carried out is of the “Find Next” type in the current method.

Find/Replace

The **Replace** command displays the following dialog box:



- The “Find What:” area is used to define the character string or the expression to be searched for. As in the Find dialog box, this area is a combo-box that stores the last 15 character strings searched for. If you highlight text before choosing the **Replace** command, it will appear in this area.
- The “Replace with:” area is used to define the character string that will replace the one defined above. This area is also a combo-box storing the last 15 character strings that have been searched for or replaced.
- The **Whole Word** option is used to find/replace only character strings that correspond exactly to the string entered. In this case, for instance, a search for “client” will not find the strings “clients” or “myclient,” etc.
- The **Case Sensitive** option is used to find/replace only character strings having the same case as that of the entered string. For instance, a search for “MyVar” will not find “myVar.”
- As in the Find dialog box, the **Previous** and **Next** buttons are used to set the direction of the search: towards the beginning or end of the current method, starting from the initial location of the cursor.

The **Replace** button is used to launch the search and replace the first occurrence found. 4D begins searching from the current text insertion point and continues to the end of the method. It is then possible to continue finding/replacing using the **Replace Next** and **Replace Previous** commands of the **Edit** menu.

The **Everywhere** button is used to replace all the occurrences corresponding to the search criteria directly in the open method.

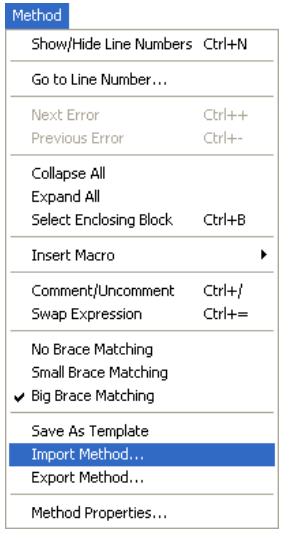
Go to Line Number

This specific search command is located in the **Method** menu. It opens a dialog box where you can indicate the line number you want to find. When you click **OK**, the editor finds and highlights that line in the method. This type of search is useful when used in conjunction with the compiler, which flags runtime errors by the line number in which they occur.

Note You can choose whether or not to display lines numbers in the Method editor window (see the "Display of Line Numbers" paragraph on page 789).

Importing and Exporting Methods

4D allows importing and exporting, in the form of a file, of database, project and object methods, as well as triggers. These commands are found in the **Method** menu:



When you select the **Export Method...** command, a standard file saving dialog box appears, allowing you to choose the name, location and format of the export file (see below).
As with printing, exporting does not take the collapsed state of code structures into account and the entire code is exported.

When you select the **Import Method...** command, a standard file opening dialog box appears, allowing you to designate the file to be imported.
Importing replaces the selected text in the method. To replace an existing method by an imported method, select the entire contents of the method before carrying out the import.

The import/export function is multi-platform: a method exported under Mac OS can be imported under Windows and vice versa; 4D handles the conversion of characters when necessary.

File Formats

4D can export and import methods in two formats:

- **4D method** (extension ".c4d" under Windows): In this format, methods are exported in encoded form. The names of objects are tokenized. This format is used in particular for exchanging methods between 4D applications and plug-ins in different languages. Conversely, it is not possible to display them in a text editor.
- **Text** (extension ".txt" under Windows): In this format, methods are exported in text-only form.
In this case, the methods are readable using a standard text editor. However, the languages of the 4D applications used for export and import must be identical.

For example, the following method, exported by 4D in Text mode...

```
C_STRING(10;$1)

If (Count parameters=0)
    <>pr_Document:=New process("List_Documents";64*1024;"List of Documents";"Open";*)
    SHOW PROCESS(<>pr_Document)
    BRING TO FRONT(<>pr_Document)
Else

    If ($1="Open")
        MENU BAR(2)
        message_for_document:=""
        page_document:=Load list("od_Page_Document")

        COPY ARRAY(<>Document_types;Document_types)

        ALL RECORDS([od_Document])

        INPUT FORM([od_Document];"Input")
        OUTPUT FORM([od_Document];"List")
        $ref:=Open form window([od_Document];"Input")

        MODIFY SELECTION([od_Document];*)
        CLOSE WINDOW
        <>pr_Document:=0
    End if

    If (Is a list(page_document))
        CLEAR LIST(page_document;*)
    End if
End case
```

... will appear as follows in a text editor:

```
C_STRING(10;$1)

If (Count parameters=0)
<>pr_Document:=New process("List_Documents";64*1024;"List of Documents";"Open";*)
SHOW PROCESS(<>pr_Document)
BRING TO FRONT(<>pr_Document)
Else

If ($1="Open")
MENU BAR(2)
message_for_document:=""
page_document:=Load list("od_Page_Document")

COPY ARRAY(<>Document_types;Document_types)

ALL RECORDS([od_Document])

INPUT FORM([od_Document];"Input")
OUTPUT FORM([od_Document];"List")
$ref:=Open form window([od_Document];"Input")

MODIFY SELECTION([od_Document];*)
CLOSE WINDOW
<>pr_Document:=0
End if

If (Is a list(page_document))
CLEAR LIST(page_document;*)
End if
End case
```

Executing Methods

Project methods written in your application are usually called automatically during the use of the application via menu commands, buttons, other methods, and so on. As for database methods, they are executed in relation to specific events that occur in the application.

However, for testing and debugging purposes, 4D lets you manually execute project methods and certain database methods in Design mode. In this case, it is possible to run the method in a new process and/or directly in Debug mode, in order to check its execution step by step

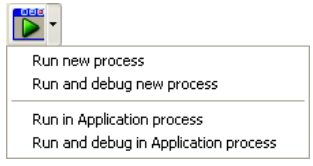
Moreover, with 4D Server, you can indicate whether 4D Server should execute a project method on the server machine or on other clients' machines.

You can execute methods in two ways:

- From the Method editor window,
- From the Execute Method dialog box (project methods only).

From the Method Editor

Each Method editor window has a button that can be used to run the current method. Using the menu associated with this button, you can choose the type of execution desired:



This button is only active for project methods and for the following database methods:

- On Startup
- On Exit
- On Server Startup
- On Server Shutdown.

The following execution modes are available:

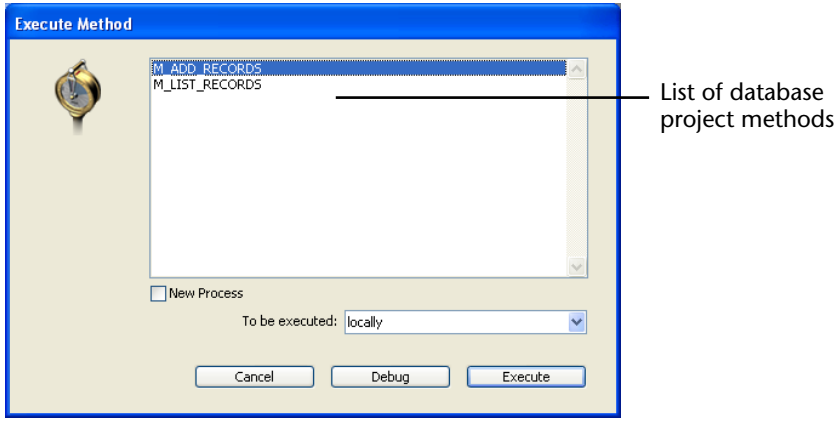
- **Run new process**: Creates a process and runs the method in standard mode in this process.

- **Run and debug new process**: Creates a new process and displays the method in the Debugger window for step by step execution in this process.
- **Run in Application process**: Runs the method in standard mode in the context of the Application process (in other words, the record display window).
- **Run and debug in Application process**: Displays the method in the Debugger window for step by step execution in the context of the Application process (in other words, the record display window).

For more information about processes, please refer to the chapter "Processes" on page 929.

From the Execute Method Dialog Box

The **Method...** command of the **Run** menu lets you select and execute an existing project method. When you select this command, 4D displays the Execute Method dialog box:



This dialog box lists all the project methods of the database, including shared project methods of components. On the other hand, project methods that have been declared invisible will not appear.

To execute a project method, simply select its name in the list and click on **Execute**. To run a method step by step in Debug mode, click on **Debug**. For more information about the 4D debugger, please refer to the *Language Reference* manual.

4D Server The **Debug** option is not available if you execute the method on the server.

If you check the **New Process** check box, the method you selected executes in another process. If the method is performing a time-consuming task such as printing a large set of records, you can continue to work with your database, adding records to a table, creating a graph to display data, and so on. For more information about processes, please refer to the chapter “Processes” on page 929.

4D Server

• If you want the method to be executed on the server machine rather than on the client machine, select the **On 4D Server** option in the To be executed menu. In this case, a new process, call the stored procedure, is created on the server machine in order to execute the method.
This option can be used to reduce network traffic and optimize the functioning of 4D Server, in particular for methods that call data stored on the disk. All types of methods can be executed on the server machine or on another client machine, except for those that modify the user interface. In this case, stored procedures are ineffective.
• You can also choose to run the method on another client workstation. Other client workstations will not appear in the menu, unless they have been previously “registered” (for more information, please refer to the description of the REGISTER CLIENT routine in the 4D *Language Reference* manual). For more information about these options, please refer to the *Language Reference* manual of 4D and to the *4D Server Reference Guide*.
By default, the **locally** option is selected. With the 4D single-user version, this is the only option available.

12 Users and Groups

If more than one person uses a database, you may want to control access to the database or provide different capabilities and interfaces to different users. If you are designing applications for use in a multi-user environment or the World-Wide Web, it may be essential that you provide security for sensitive data. You can provide this security by assigning passwords to users and creating access groups that have different levels of access to information in the database or to database operations.

This chapter provides information about 4D's password access system. You use this system to:

- Specify the users of a database,
- Provide users with passwords,
- Create groups of users with different levels of access to the database,
- Nest groups of users within other groups to create a hierarchy of users,
- Set a default user,
- Specify the group which owns the objects each user creates,
- Assign a startup method for each user,
- Specify a group owner.

After you create access groups, you can manage access to:

- The Design Environment
- Record operations
- The SQL server,
- Forms
- Methods
- Menu commands
- Plug-ins.

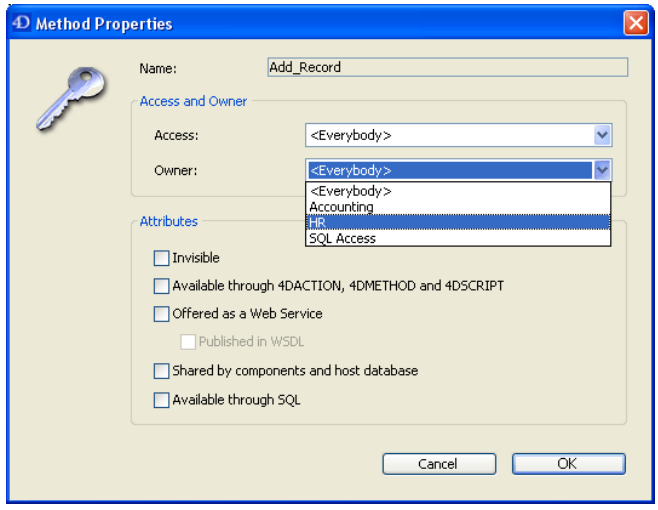
In addition to providing security for your databases, the access system also maintains a user history — the Users editor can tell you how many times a user has accessed the database and the date of the most recent access.

4D Server Object locking occurs when two or more users attempt to modify the password access system at the same time. Only one user can use the Users and Groups editor at a time.

Access System Overview

4D's password access system is based on users and groups. You create users and assign passwords, put users in groups, and assign each group access rights to appropriate parts of the database. Groups can be assigned access privileges to operations on records in the table and to the table definition.

The following example shows “Owner” access rights for the *Add_Record* project method being assigned to a group. Groups can generally be assigned “Access” (use) and/or “Owner” (modification) access rights.

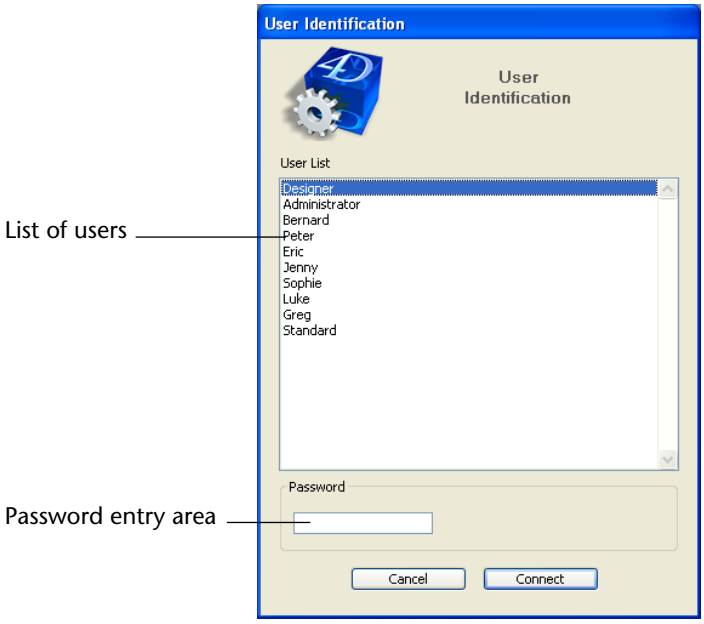


To open the database, a user either selects or types his or her user name and then types his or her password[1]. Then, depending on which groups the user belongs to and to which parts of the database the groups have been assigned, the user can operate the parts of the database that were specified by the access system.

The **Access** page (“Application” theme) of the Preferences dialog box lets you set the access mode of the database (see the “Access Page” paragraph on page 166).

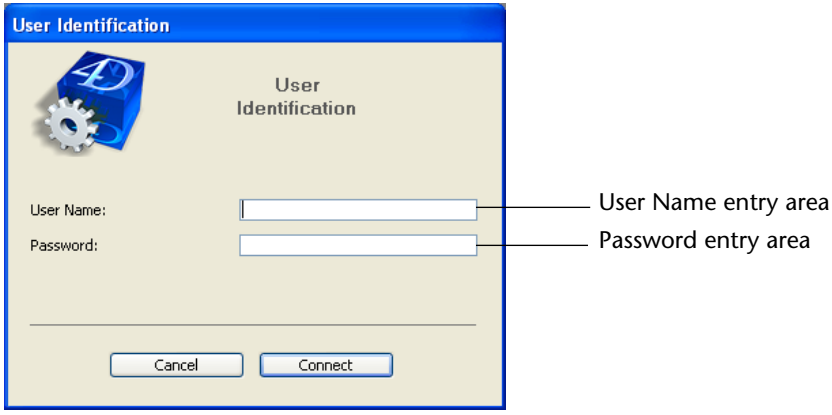
1. Unless the **Default User** option is used (for more information, refer to “Setting a Default User” on page 841).

By default, the following password entry dialog box is displayed.



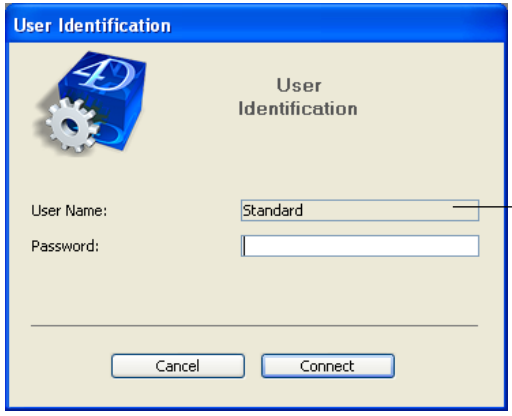
In this dialog box, the user selects his or her name from the list of users and types his or her password in the password entry area.

If you deselect **Display User List in Password Dialog Box** in the Preferences dialog box, the password entry dialog box shown below will be displayed.



In this dialog box, the user must type both his or her name and password, which reinforces application security.

If you have set a Default User in the Preferences dialog box and have assigned it a password, the following dialog box is displayed:



Users only have to enter the password. If no password is assigned to the Default User, the dialog box is not displayed. Each user, in this case, has the same privileges and restrictions as set for the Default User.

If the "The user can change their password" option is checked in the application Preferences (see the "User Access" paragraph on page 168), the **Change** button is displayed in the password entry dialog box. This button lets the current user modify his or her own password.

4D Server After logging in to the database using any of the above dialog boxes, the user may choose to save the location of the server database (the pathname) and (optionally) the password used to log in to the database. The next time that the user double-clicks the 4D Client application icon, the database is automatically run and, if the user has saved his or her password, the user is automatically logged on to the database. For more information, refer to the *4D Server Reference* manual.

The user operates the database in a normal fashion. When the user attempts to use an object (form, menu command, method) that its group is not permitted to use, 4D displays an error message of the type "Your password does not allow you to use this object."

Note If an *ON ERR CALL* method is installed, the error message for methods is not displayed. Refer to the *4D Language Reference* manual for more information.

External Access Using 4D Password System

Once specified, the access control system can be used for several types of external access to the 4D database. You can therefore take advantage of the access hierarchy in these specific contexts.

The users and groups system of 4D can be used for:

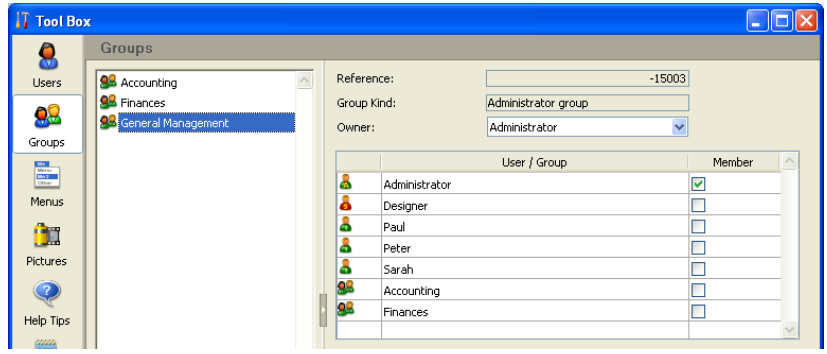
- The **integrated Web server of 4D**. For more information about how 4D users and groups are taken into account during connections to the 4D Web server, refer to the "Connection Security" section of the *Language Reference* manual.
- The **integrated SQL server of 4D**. For more information about access groups for the integrated SQL server of 4D, please refer to the "SQL Server Access" paragraph on page 235 and to the *4D SQL Reference* manual.
- **4D Open access**. For more information about this point, please refer to the "4D Open" paragraph on page 216.

An Access Hierarchy Scheme

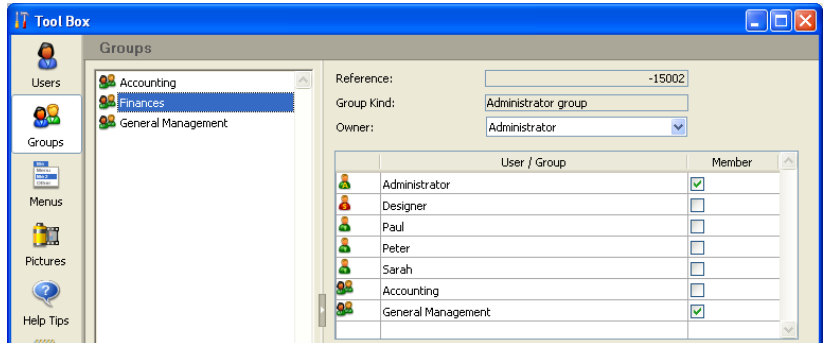
The best way to ensure the security of your database and provide users with different levels of access is to use an access hierarchy scheme. Users can be assigned to appropriate groups and groups can be nested to create a hierarchy of access rights. This section discusses several approaches to such a scheme.

In this example, a user is assigned to one of three groups depending on their level of responsibility. Users assigned to the Accounting group are responsible for data entry. Users assigned to the Finances group are responsible for maintaining the data, including updating records and deleting outdated records. Users assigned to the General Management group are responsible for analyzing the data, including performing searches and printing analytical reports.

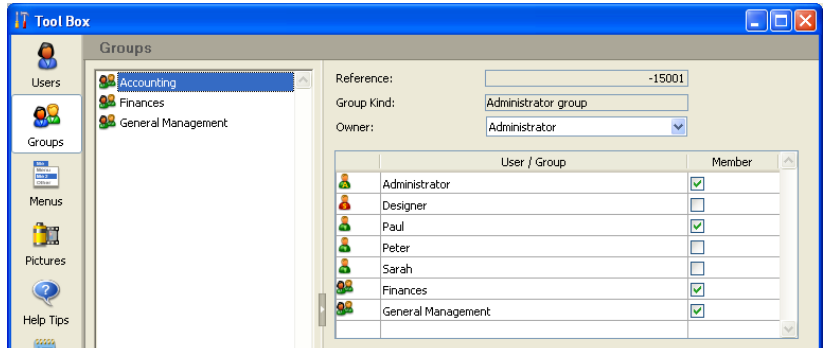
The groups are then nested so that privileges are correctly distributed to the users of each group. The General Management group contains only “high-level” users.



The Finances group contains data maintenance users as well as General Management users, thus the users in General Management have the privileges of the Finances group as well.



The Accounting group contains data entry users as well as Finances group users, so the users who belong to the Finances group and the General Management group enjoy the privileges of the Accounting group as well.



You can decide which access privileges to assign to each group based on the level of responsibility of the users it includes. If you assign the Accounting group to an input form, for example, it means that everyone can use this input form. If you assign the Finances group to the form, it is restricted to members of the Finances and General Management group. If you assign the General Management group, only members of this group can use the form.

Such a hierarchical system makes it easy to remember to which group a new user should be assigned. You only have to assign each user to one group and use the hierarchy of groups to determine access.

Your access scheme should restrict access at the lowest possible level, usually at the form level.

The Designer and the Administrator

4D provides users with certain standard access privileges and certain powers. Once a users and groups system has been initiated, these standard privileges take effect.

The most powerful user is named Designer. The Designer has control over the design of the database. The Designer can create users and groups, assign access privileges to groups, and use the Design environment. No aspect of the database is closed to the Designer.

After the Designer, the next most powerful user is the Administrator, who is usually given the task of managing the access system. When the Users page of the tool box is first opened, both the Designer and Administrator appear in the list of users. At this point, the Administrator is just a regular user with no special access privileges, in particular if access to the Design mode is restricted. To be able to use the access system, the Administrator must be given special access privileges. For information about this, see the section “Administrator and Group Owner Access” on page 856. The Administrator is the only user with the ability to save and load groups. For information about saving and loading groups as the Administrator, see the section “Loading and Saving Groups” on page 851.

The Administrator's access to other parts of the database is limited by group membership — the Administrator must be part of one or more groups to have access privileges in the database. The Administrator is placed in every new group, but you can remove the Administrator's name from any group.

In the user management dialog box, the icons of the Designer and Administrator are displayed in red and green respectively:

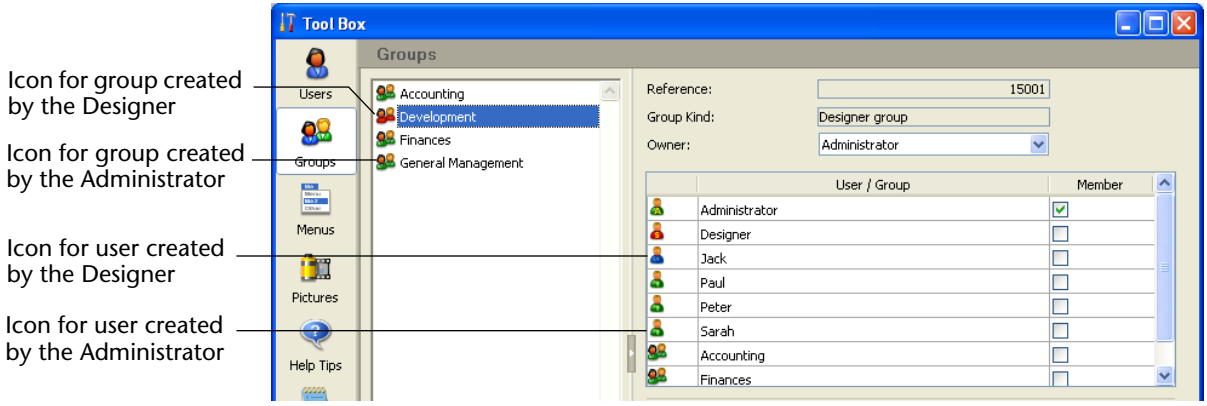
 — Designer icon

 — Administrator icon

You can rename the Designer and Administrator users but their icons cannot be changed.

You can distinguish between users and groups created by the Designer and Administrator by the color of their icons:

- Icons for groups created by the Designer are red and those created by the Administrator are green.
- Icons for users created by the Designer are blue whereas those created by the Administrator are green.



The group owner can change the default name at any time.

The Designer and Administrator can each create up to 16,000 groups and 16,000 users.

Group Owners

You can designate an owner for each group. Usually, the owner is the Administrator, but you can designate any group member as the owner.

The group owner can be given the ability to add and remove users from any group he or she owns. The users to be added must already exist. Group owners cannot create users, or change user properties such as passwords. Group owners cannot add or remove other groups.

As with the Administrator, it may be necessary to grant access to the password editor explicitly for the group owner when access to the Design mode is restricted. For information about this, see the section "Administrator and Group Owner Access" on page 856.

Giving Users Design Environment Access

All users have access to the Application mode. However, you can restrict access to the Design mode. To do this, you simply need to select a group from the **Design Access** drop-down list on the "Access" page of the Preferences dialog box. For more information, see the "Access Page" paragraph on page 166.

In this case, only users belonging to this group as well as the Designer can modify the database structure. The Designer always has access to the Design environment, even if they do not explicitly belong to the design access group.

All other users are ordinary users. When a user opens the database, it opens in the Application environment. The access of a user is limited by their group membership.

Activating the Password System

You initiate the 4D password access system by assigning a password to the Designer.

Until you give the Designer a password, 4D allows anyone to use any part of the database.

When a password is assigned to the Designer, all the access privileges you have assigned to tables, forms, menus, and methods take effect. In order to open the database, users must enter a password.

Important ***Do not forget the Designer's password! If you do, you will be unable to open the database in the Design environment.***

To disable the access system, you just need to remove the Designer password.

Setting a Default User

You can set a Default User to use your database. When this option is active, users that open or connect to the database are no longer required to enter a name. Moreover, if you have not associated a password with the Default User, the password entry dialog box does not appear and the database opens directly.

Once logged on as a Default User, each user has the access privileges and restrictions set for the Default User.

This option simplifies access to the database while maintaining a complete control system for user actions.

▶ To set a Default User:

1 **In the Design environment, create a user (the name you choose is not important) in the Password editor.**

You can associate a password with the user, but it is not mandatory. For more information, refer to "Managing Users and Groups" on page 843.

2 **Using the Design environment editors, choose the access privileges and restrictions for this user.**

For more information, refer to "Assigning a Group To Database Objects" on page 852.

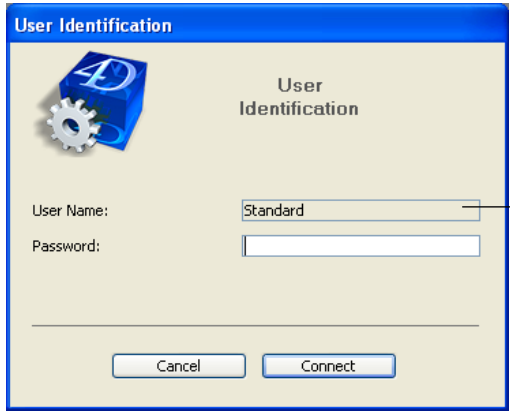
3 **In the Preferences window, go to the Access page ("Application" theme).**

For more information, refer to the "Access Page" paragraph on page 166.

4 **Choose your user in the Default User drop-down list and validate the dialog box.**

The access to the database is now no longer customized.

- If you have not associated a password with the Default User, the dialog box does not appear.
- If you have associated a password with the Default User, a dialog box appears and the Default User's password must be entered:



Note When the Default User mode is activated and a password is required, it is recommended to deselect the "The user can change their password" option on the "Access" page of the Preferences.

Redisplaying the Password Dialog Box

You can force 4D to display the standard password entry dialog box in order, for example, to connect to the database as the Designer or Administrator.

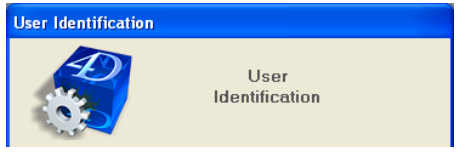
► To redisplay the password entry dialog box when the Default User mode is active:

1 **Open the database while holding down the Shift key.**

A password entry dialog box appears allowing you to enter a name and password.

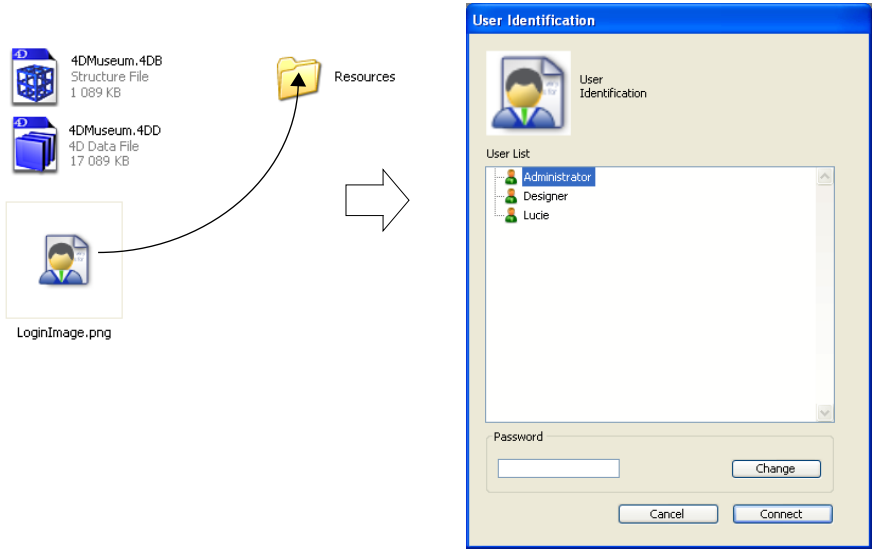
Customizing Icon of ID Window

It is possible to customize the icon displayed in the database connection dialog box. By default, this icon depicts the 4D logo.



To replace this icon by one of your choice, you simply need to place a file named **LoginImage.png** in the **Resources** folder of the database (located next to the database structure file, see the "Database Architecture" paragraph on page 42).
The custom file must be of the "png" type and its size must be 80x80 pixels.

When the database is opened, 4D will load this picture instead of the default one:



Managing Users and Groups

The editors for users and groups are located in the tool box of 4D. These editors can be used to create both users and groups, assign passwords to users, place users in groups, etc.

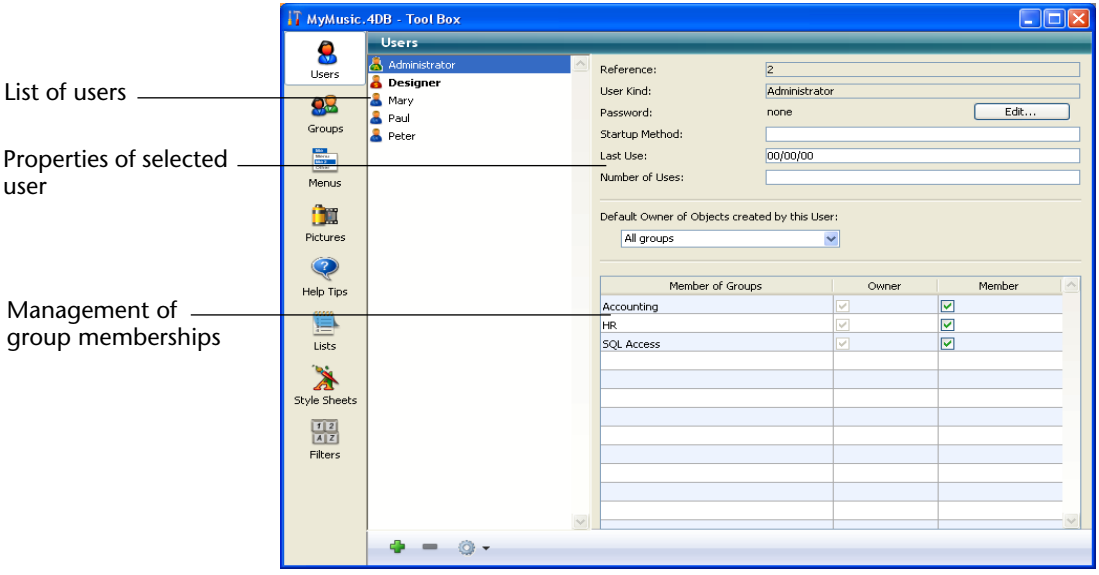
Adding and Modifying Users

You use the users editor to create user accounts, set their properties and assign them to various groups, in addition to monitoring their use of the database.

► To add a user:

1 Select Tool Box > Users from the Design menu or click on the "Tool Box" button of the 4D tool bar.

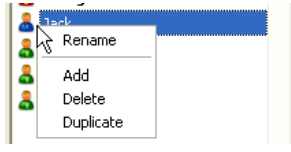
4D displays the users editor.



The list of users displays all the users “visible” by the current user, i.e.:

- For the Designer: All users,
- For the Administrator: Users that they have created (green icons).

2 Click on the add button located below the list of users.
OR
Right-click in the list of users and choose the Add or Duplicate command in the contextual menu.



Note The **Duplicate** command can be used to create several users having the same characteristics quickly.

4D adds a new user to the list, named *New userX* by default.

The properties area displays information about the user type:

- The “Reference” field indicates the reference number of the selected user. This number is used by the language commands.

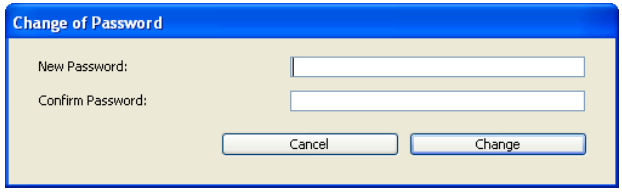
- The “User Kind” field indicates from where the user profile comes. The user types are as follows:
  - Designer: Designer user
  - Administrator: Administrator user
  - Developer: User created by Designer
  - User: User created by Administrator.

3 **Enter the new user name.**

This name will be used by the user to open the database.
You can rename a user at any time using the **Rename** command of the contextual menu, or by using the **Alt+click** (Windows) or **Option+click** (Mac OS) shortcuts, or by clicking twice on the name you want to change.

4 **Enter the password for the user by clicking the Edit... button in the user properties area.**

The following dialog box appears:



5 **Type the password in the New Password entry area and enter it again in the Confirm Password area.**

You can use up to 15 alphanumeric characters for a password. The password editor is case sensitive — the user must enter the password exactly as it is entered here. For example, if you define a user’s password as “HolyCow,” the user must enter it with a capital H and capital C; otherwise 4D will not accept it.
When a password is entered, it is not visible in the dialog box. Asterisks are displayed instead of each character entered.

6 **Validate the dialog box.**

If the two password entries are different, 4D plays a Beep and cancels the password modification.

7 **Choose a group from the "Default Owner of Objects created by this User" drop-down list.**

This group owns any objects (forms, methods, and so on) that the user creates. For instance, you might specify that the Accounting group owns the objects created by each user in the Accounting group. If a user from another group attempts to modify a form created by a member of the Accounting group, a message appears stating that the user does not have adequate privileges to edit the form.

8 **Enter the name of an associated method that will be executed when the user opens the database (optional).**

This method can be used for example to load the user preferences.

9 **Set the group(s) to which the user belongs using the "Member of Groups" table.**

- You can add the selected user to a group by checking the corresponding option in the **Member** column. You can also remove the user from a group by unchecking this same option.
- The **Owner** column indicates whether the selected user is a group owner. This column cannot be modified.

Note The membership of users to different groups can also be set by group on the **Groups** page.

To modify the characteristics of an existing user, simply select the user in the list then carry out the modifications. Refer to steps 3 to 9 above for more information about user parameters.

Deleting a User

To delete a user, select it then click the deletion button or use the **Delete** command of the contextual menu.

It is not possible to physically delete a user account. When you ask to delete an account, 4D requests that you confirm the operation and indicates that the account will appear as deleted in the list of users. If you validate the dialog box, the account is disabled and can no longer be used. A deleted user appears dimmed in the list.

This is necessary in order to guarantee the uniqueness of user reference numbers.

Creating Access Groups

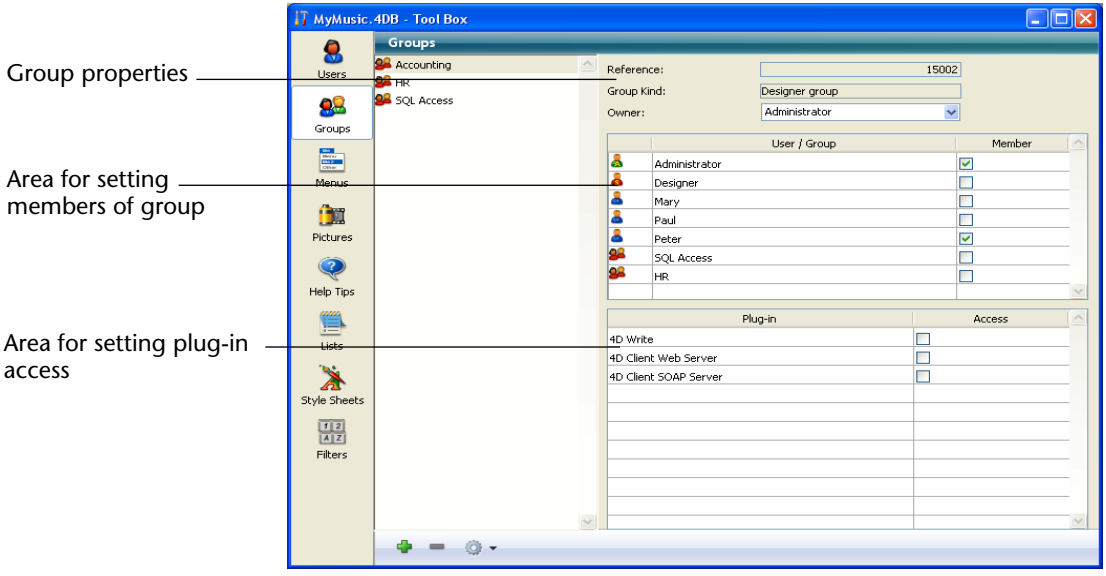
You can use the groups editor to set the elements that each group contains (users and/or other groups) and to distribute access to plug-ins. When you create a group, you can designate its owner from among the users.

Keep in mind that once a group has been created, it cannot be deleted. If you want to deactivate a group, you just need to remove any users it contains.

► To create a group:

1 **Select Tool Box > User groups in the Design menu or click on the "Tool Box" button of the 4D tool bar then on the Groups button.**

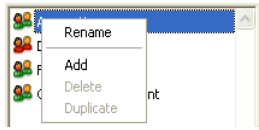
4D displays the groups editor window:



The list of groups displays all the groups of the database, regardless of which user created them.

2 **Click on the add button located below the list of groups.**
OR

Right-click in the list of groups and choose the Add or Duplicate command in the contextual menu.



Note The **Duplicate** command can be used to create several groups having the same characteristics quickly.

4D adds a new group to the list, named *New groupX* by default.

The properties area displays information about the group:

- The “Reference” field indicates the reference number of the selected group. This number is used by the language commands.
- The “Group Kind” field indicates from where the group comes. The group types are as follows:
  - Designer group: Group created by the Designer.
  - Administrator group: Group created by the Administrator .

3 **Enter the name of the new group.**

The group name can be up to 15 characters long.

You can rename a group at any time using the **Rename** command of the contextual menu, or by using the **Alt+click** (Windows) or **Option+click** (Mac OS) shortcuts, or by clicking twice on the name you want to change.

You can only rename groups that you have created. The Designer cannot rename a group created by the Administrator and vice versa.

4 **Select an owner from the “Owner” drop-down list (optional).**

The group owner can add and remove users from the group (see the “Group Owners” paragraph on page 840). Keep in mind that by default, the Administrator is owner of all the groups, even those created by the Designer.

5 **Set the members of the group by checking the corresponding options in the Member area.**

6 **Distribute access to plug-ins (optional).**

For these last two steps, refer to the following paragraphs.

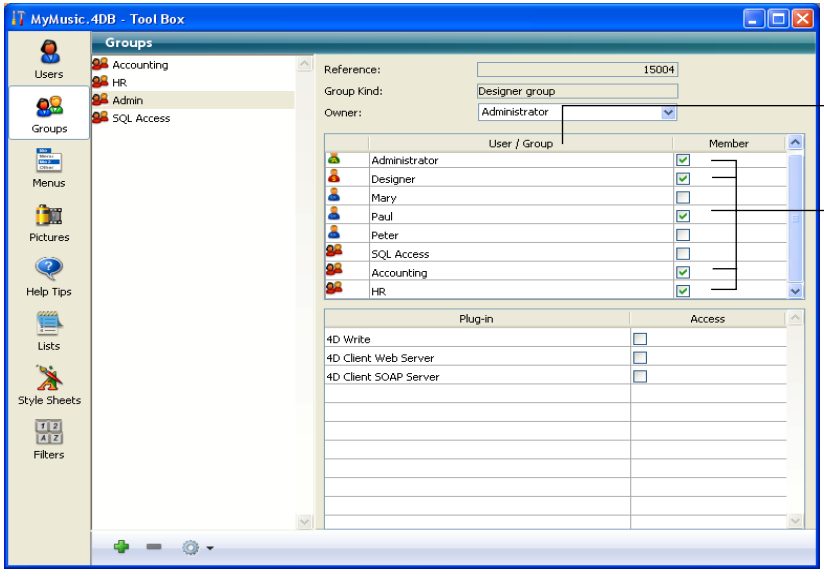
Placing Users or Groups into Groups

You can place any user or group into a group, and you can also place the group itself into several other groups. It is not mandatory to place a user in a group.

The Designer can modify the contents of any group in the database. The Administrator and the users that are group owners can only modify the groups for which they are owners.

However, regardless of your user status, you can view, add or remove any user or group from your own groups.

To place a user or group in a group, you simply need to check the corresponding option in the member attribution area.



If you check the name of a user, this user is added to the group. If you check the name of a group, all the users of the group are added to the new group.

The affiliated user or group will then have the same access privileges as those assigned to the new group.

Placing groups into other groups lets you create a user hierarchy. The users of a group placed in another group will have the access privileges of both groups. For example, if you place the General Management group in the Accounting group, the users of the General Management group will benefit from the privileges of both groups; however, users which are only placed in the Accounting group will benefit from the privileges of that group only. For more explanations about the operation of an access system hierarchy, refer to the section "An Access Hierarchy Scheme" on page 836.

To remove a user or group from another group, you just need to deselect the corresponding option in the member attribution area.

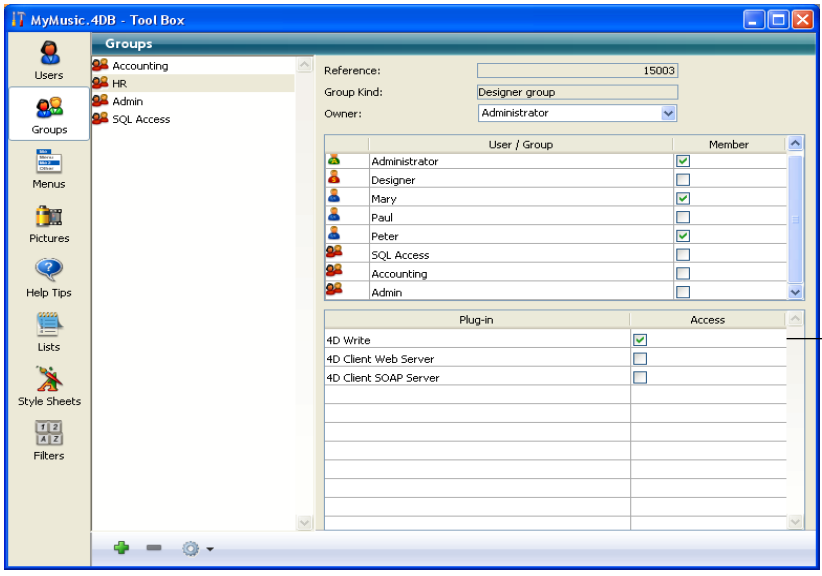
Assigning a Group to a Plug-in or to a 4D Client Web Server

You can assign a group privileges to any plug-ins installed in the database. This includes all the 4D plug-ins and any third-party plug-ins. For more information about plug-ins, refer to the section "Plug-in Areas" on page 592.

Distributing access to the plug-ins lets you control the use of the licenses you possess for these plug-ins. Any users that do not belong to the access group of a plug-in cannot load this plug-in.

You can also restrict the use of the 4D Client Web server and SOAP server via the plug-in access area.

The “Plug-in” area on the **Groups** page of the tool box lists all the plug-ins loaded by the 4D application. To give a group access to a plug-in, you simply need to check the corresponding option:



Members of the HR group have access to the 4D Write plug-in

The **4D Client Web Server** and **4D Client SOAP Server** items lets you control the possibility of Web and SOAP (Web Services) publication for each 4D Client. These licenses are considered as plug-in licenses by 4D Server. Therefore, in the same way as for plug-ins, you can restrict the right to use these licenses to a specific group of users.

Loading and Saving Groups

4D allows the Administrator to save and load any groups that he or she has created or modified. When groups are saved, everything about the current users and groups are saved.

The ability to save groups means that the Administrator can save the access system of a database and transfer it to a modified version of the same database or to a new database. This is extremely useful for restoring the access system for a new version of the database. Because the groups can be reloaded, users of the database do not have to learn a new access system.

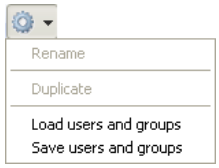
All the user names, passwords, startup method names, groups, group owners, and group memberships are preserved.

Note The Designer cannot save or load groups via the Groups editor.

► To save groups created or modified by the Administrator:

1 **Enter the database as the Administrator and display the Groups page of the tool box.**

2 **Click on the options menu and choose the Save users and groups command.**



4D displays a file creation dialog box so that you can name and save the group.

Note Group and user files have the ".4UG" extension.

► To load groups:

1 **Enter the database as the Administrator and display the Groups page of the tool box.**

2 **Click on the options menu and choose the Load users and groups command.**

4D displays a dialog box so that you can select the group file.

Assigning a Group To Database Objects

After you set users and access groups, you can assign groups to the following objects:

- Forms,
- Methods,
- Menu commands.

You may assign only one group to each object. For this reason, it is important to design the access groups so that more powerful users belong to all the groups below them in the access hierarchy.

For a discussion of how to organize users and access groups, see the section "An Access Hierarchy Scheme" on page 836.

Assigning a Group to a Form

When you assign a group access privileges to a form, only users belonging to the group can use that form for data entry. When you assign a group owner privileges to a form, only users belonging to the group can modify that form in the Design environment.

► To assign access and owner privileges:

1 **Open the form in the Form editor then display its properties in the Property List.**

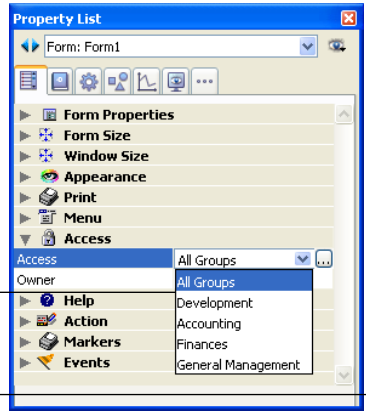
For more information, refer to "The Property List" paragraph on page 388.

OR

Select the form on the Forms page of the Explorer then select the Form Properties... command from the contextual menu or options menu.

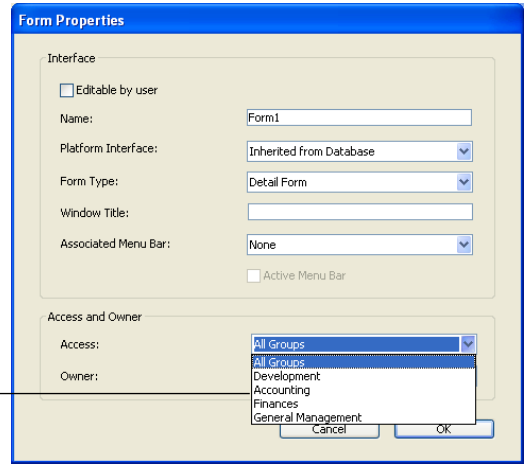
Both the Property List and the "Form Properties" window include Access and Owner lists:

Property List



Drop-down lists for assigning Access and Owner groups

Form Properties window



2 **Choose a group from the "Access" drop-down list.**

If you do not choose a group, all groups retain access privileges for the form (default setting).

3 **Choose a group from the "Owner" drop-down list.**

If you do not choose a group, all groups retain owner privileges for the form (default setting).

Assigning a Group to a Project Method

When you assign a group access to a project method, only users belonging to the group can use that method. When you assign a group ownership of a project method, only users belonging to the group can modify that method in the Design environment.

► To assign access and owner privileges:

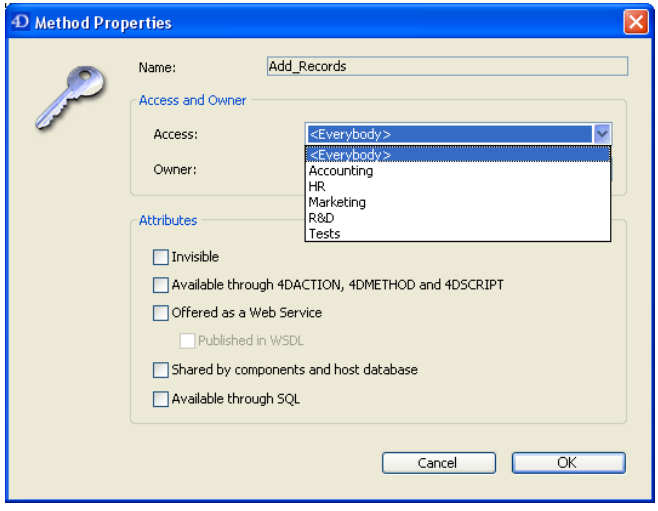
1 **Open the method in the Method editor then select the Method Properties... command in the Method menu.**

For more information, refer to the "Creating or Opening a Project Method" paragraph on page 766.

OR

Select a project method on the Methods page of the Explorer then select the Method Properties... command in the contextual or options menu.

The Method Properties dialog box includes the Access and Owner lists:



2 **Choose the group from the "Access" drop-down list.**

If you do not choose a group, all groups retain access privileges for the method (default setting).

3 **Choose a group from the "Owner" drop-down list.**

If you do not choose a group, all groups retain owner privileges for the method (default setting).

4 **Click the OK button.**

Assigning a Group to a Menu Command

You can assign an access group to a menu command so that only users in that group can use the menu command in the Application environment.

▶ To assign an access group to a menu command:

1 **Select Tool Box > Menus from the Design menu or click on the "Tool Box" button of the 4D tool bar then click on the Menus button.**

The Menu Bar editor appears. For more information about this editor, please refer to the chapter "Custom Menus" on page 859.

2 **Select a menu bar.**

The central list area shows the menus belonging to this menu bar.

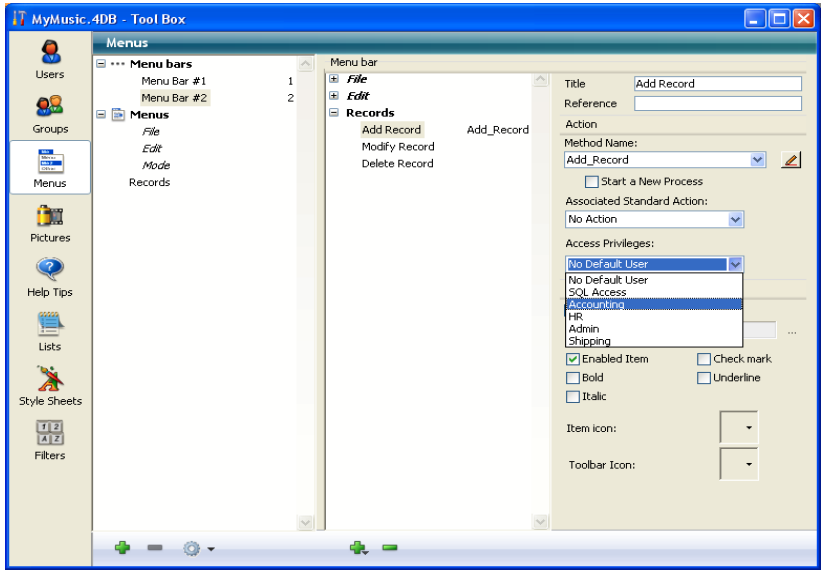
3 **Expand a menu in the list of menu bars.**

The menu commands and methods for the selected menu appear.

4 **Select the menu command for which you want to specify an access group.**

5 **Select the group from the "Access Privileges" drop-down list.**

The figure below shows the Accounting group being given access privileges to the **Add Record** menu command.



System Maintenance

Once a password access system is in place, occasional maintenance of the system is necessary. New users must be added, groups memberships may need to be modified, and passwords need to be changed regularly. The Designer has access to the Design environment and can make any necessary modifications using the Users and Groups pages of the tool box.

The Administrator and the Designer can also view the usage history of each user as necessary for maintenance.

Administrator and Group Owner Access

The Administrator does not necessarily have access to the Design environment. However, if the Designer creates a project method that contains the EDIT ACCESS command, the Administrator and group owners can have limited power to control users and groups.

The EDIT ACCESS command can be included in a method that is attached to a custom menu, or can be executed by choosing the **Run>Method...** menu command. If the method is executed by a user who is not the Administrator or another group owner, it has no effect.

When the method is executed, the result depends on whether the user is the Administrator or a group owner.

If the Administrator executes the method that contains the EDIT ACCESS command, 4D displays the tool box containing only the users and groups editors. The Administrator can use these editors to create users and groups; edit any users or groups he or she created, including changing user passwords; and add or remove users from any groups he or she created.

The Administrator cannot assign groups to forms, menu commands, methods, or plug-ins. Only the Designer can assign these access groups.

If a group owner who is not the Administrator executes the method that contains the EDIT ACCESS command, 4D displays the tool box containing only the groups editor. Moreover, the editor only contains groups for which the user is owner. The group owner can add or remove users from the groups. The group owner cannot create users, edit user information, or add groups. The commands for adding and editing users and groups are dimmed.

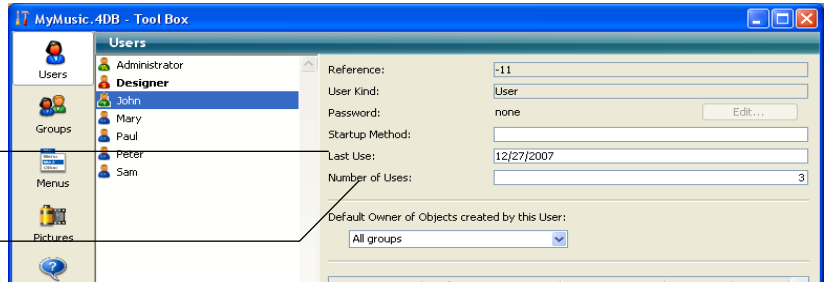
Viewing Usage

The Users page of the tool box contains the date of the user's last use of the database and the total number of uses. The Administrator or Designer can view this information by selecting a user in the list.

▶ To view the user information:

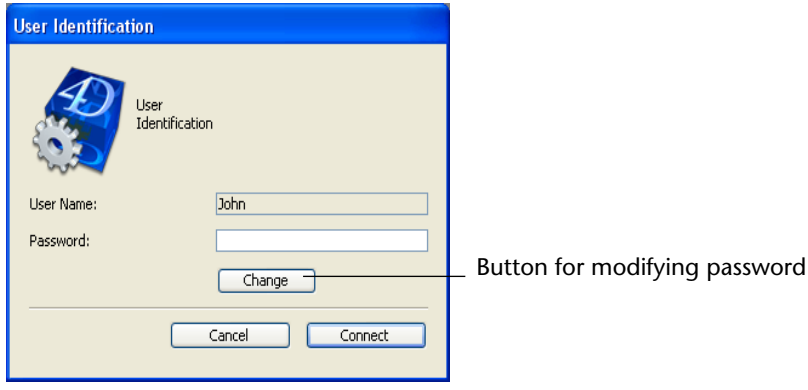
1 **Choose Tool Box > Users from the Design menu or click on the "Tool Box" button of the 4D tool bar.**
OR
Execute the method that contains the EDIT ACCESS command.
4D displays the tool box containing the users editor.

2 **Select the user name that interests you from the list of users.**
The dialog box displays the date of the user's last use of the database as well as the number of times the user has opened the database.

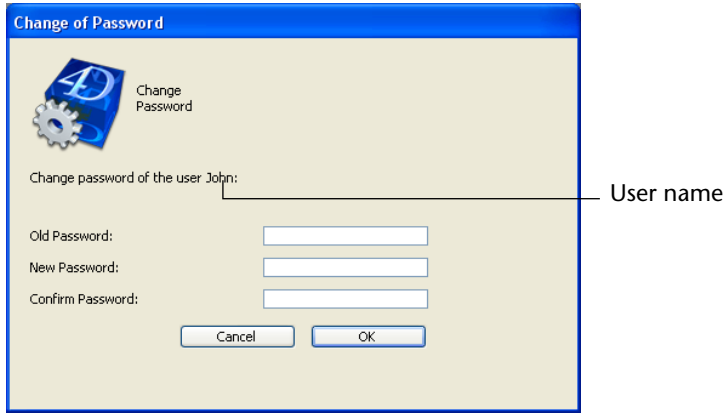


Modification of Password by User

The User Identification dialog box by default includes a **Change** button which lets the current user modify their own password. If the list of users is displayed, the current user is the selected user. If the list is not displayed, the name of the current user must be entered beforehand in the User Identification dialog box:



When the user clicks the **Change** button, the following dialog box appears:



In order to modify their password, the user must indicate their old password. The new password is then entered and confirmed. Once the dialog box is validated, if the entry is correct, the new password of the user replaces the old one and is stored in the database. The user then needs to enter his or her new password and click the **Connect** button in order to open the database.

It is possible to hide the **Change** button in order to prevent users from modifying their password. This option is found on the Application/Access page of the application Preferences (see the "User Access" paragraph on page 168).

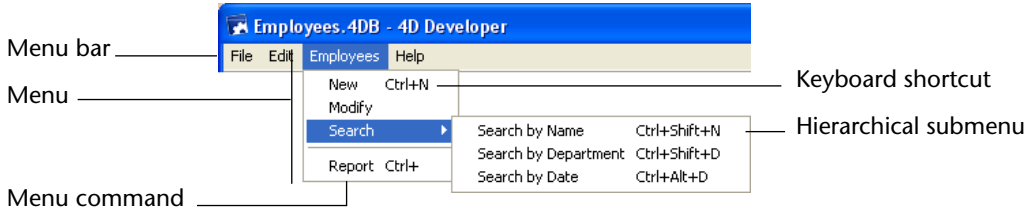
13 Custom Menus

You can create custom menus for your databases and custom applications. Because pull-down menus are a standard feature of any desktop application, their addition will make your databases easier to use and will make them feel more familiar to users. When you create custom menus, you can also create custom toolbars. With custom menus and toolbars, your databases will perform more like “stand-alone” applications.

A custom application must contain at least one menu bar with one menu. By default, when you create a new database, 4D creates a custom menu bar so that you can access the Application environment. For detailed information about creating custom applications, refer to the *4D Language Reference* manual.

Designing an Interface with Menus

In general, menus provide menu commands that the user chooses to perform database tasks: modifying records, searching for records, printing reports, and so on. The figure below shows an example of custom menus.



A menu bar is a group of menus that can be displayed on a screen together. Each menu on a menu bar can have numerous menu commands in it, including some that call cascading submenus (known as hierarchical submenus). When the user chooses a menu or submenu command, it calls a project method or a standard action that performs an operation.

You can have many separate menu bars for each database. For example, you can use one menu bar that contains menus for standard database operations and another that becomes active only for reporting. One menu bar may contain a menu with menu commands for entering records. The menu bar appearing with the input form may contain the same menu, but the menu commands are disabled because the user doesn't need them during data entry.

You can also use the Menu Bar editor to create custom toolbars. To do so, you associate an icon with a menu command. The icon appears in the 4D toolbar and the text of the menu command is used as the icon's Help Tip.

You can use the same menu in several menu bars or other menus, or you can leave it unattached and manage it only by programming (in this case, it is known as an independent menu).

When you design menus, keep the following two rules in mind:

- **Use menus for functions that are suited to menus**: Menu commands should perform tasks such as adding a record, searching for records, or printing a report.
- **Group menu commands by function**: For example, all menu commands that print reports should be in the same menu. For another example, you might have all the operations for a certain table in one menu.

You use the Menu Bar editor to create menus. You can perform the following operations in the Menu Bar editor:

- Create and rename menu bars,
- Specify styles for menu commands,
- Associate keyboard shortcuts and icons with menu items,
- Create custom tool bars,
- Place separator lines between menu commands,

- Associate a project method or a standard action with a menu command,
- Attach a menu to another menu in order to create a hierarchical submenu,
- Assign groups of users to menu commands,
- Enable or disable menu commands,
- View sample menus while you are creating the menu bar,
- Paste in a custom graphic associated with each menu bar that will be displayed as a splash screen,
- Specify that a new process start when a menu command is chosen.

Each of these tasks is explained in this chapter.

Note that the 4D language can be used to create and manage menu bars and menus entirely by programming, without using the Menu editor. For more information about this point, please refer to the 4D *Language Reference* manual.

Creating Menus

4D allows you to create entire menu bars. A menu bar is the collection of menus that appears at the top of your application window. The menu bar displays the menu titles and the menus pull down to display the menu and submenu commands. Every menu or submenu command must be associated with a project method or a standard action.

4D Server Object locking occurs when two or more users attempt to modify the same menu bar at the same time. If a user is modifying a menu in a menu bar, the menu is locked. Other users can modify different menus in that menu bar, but they cannot modify the same menu. In addition, if a user is modifying any aspect of a menu bar, other users cannot add any new menus to the menu bar.

Basic Steps for Creating Menus

The following are the basic steps for creating custom menus using the Menu editor:

1 **Create one or more menu bars.**

See the section "Creating a Menu Bar" on page 866.

2 **Create one or more menus.**

See the sections “Creating Menus” on page 867 and “Adding Menu Commands” on page 869.

3 **Add menu commands to each menu in the menu bar.**

See the section “Attaching a Menu to a Menu Bar” on page 870.

4 **(Optional) Attach menus to other menus in order to create hierarchical submenus.**

For more information about this point, please refer to the paragraph “Attaching a Menu to Another Menu (Submenus)” on page 872.

5 **Assign a project method or a standard action to each menu command.**

When the user chooses a menu command, 4D executes the method or standard action associated with it. See the paragraph “Specifying the Action of a Menu” on page 875.

6 **Make any enhancements you want such as special font styles, separator lines, keyboard shortcuts, toolbar or menu item icons, and so on (optional).**

See the paragraph “Setting Menu Properties” on page 879.

7 **Assign password access groups to menu commands (optional).**

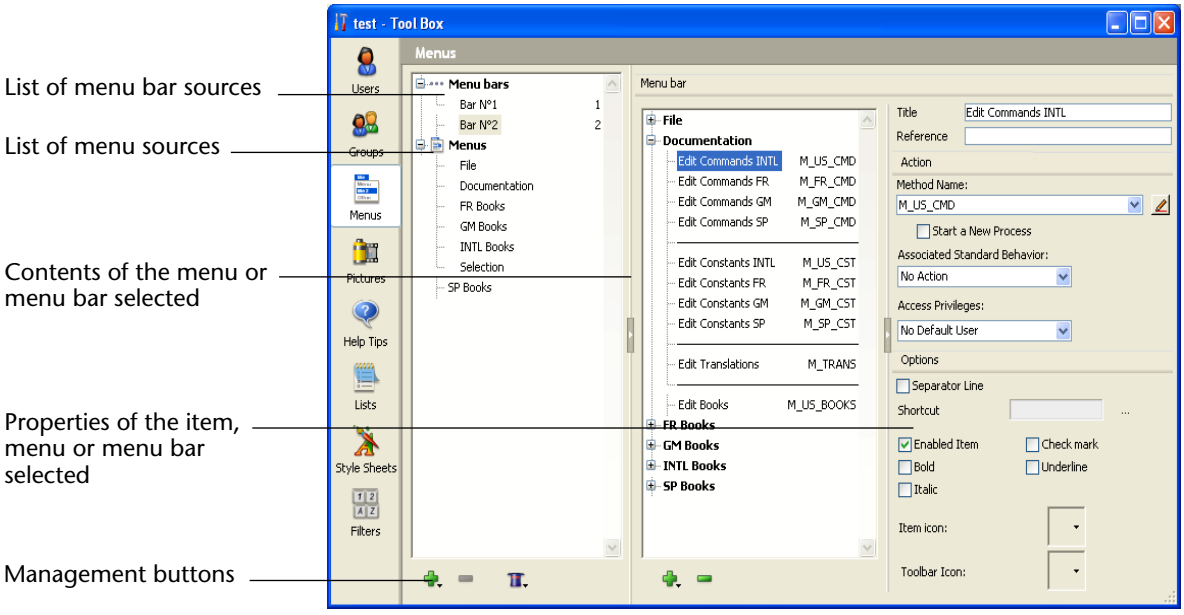
See the section “Assigning a Group To Database Objects” on page 852.

8 **Specify that a new process be started when a menu command executes (optional).**

See the section “Start a New Process” on page 877.

Menu Editor

The Menu editor can be accessed using the **Menus** button of the Tool box.

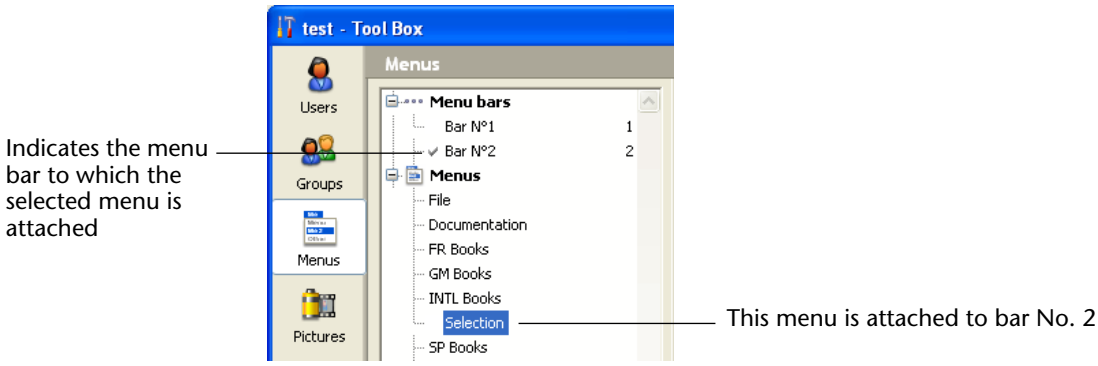


Menus and menu bars are now displayed as two items of the same hierarchical list, on the left side of the dialog box. This means that you can view all the menus defined in the database at once, without it being necessary to first select a menu bar. The menus are listed in alphabetical order.

Each menu can be attached to a menu bar or to another menu. In the second case, the menu becomes a sub-menu (see the paragraph “Attaching a Menu to Another Menu (Submenus)” on page 872).

To view the contents of a menu bar or a menu, click on its title in the left-hand list of the editor. The list of items belonging to the menu bar or menu will be displayed in the central area. The properties of the menu bar or menu will also appear on the right-hand side of the window. To display the properties of a menu item, select it in the central part of the window.

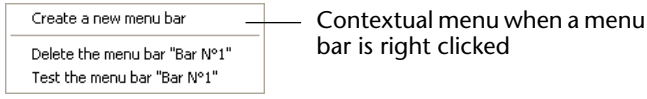
A check mark (√) indicates the element (menu bar or menu) to which the selected menu belongs. When a menu is associated with several items, several check marks will be displayed.



If the menu is not used (independent menu), no check mark appears.

Contextual Menu and Options Menu

The Menu editor includes a contextual menu, which can be used to directly access possible actions depending on the type of item that was clicked (menu bar, menu, items).



The contextual menu can be used to add or remove an item, expand or collapse the list and provide more specific actions.

The Menu editor also includes an options menu that can be accessed by clicking on the button shaped like a gear, which is found below the left-hand list. This menu includes both permanent commands and contextual commands:

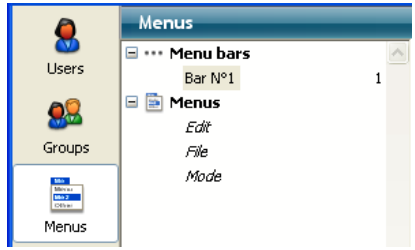
The permanent commands let you create a new menu bar or new menu, as well as a standard **Edit** menu. The contextual commands depend on the item selected (menu bar or menu) and offer appropriate management actions.

Default Menu Bars

When you create a new database, 4D automatically creates a default menu bar (Menu Bar #1) including standard menus and a command for returning to the Design mode.

This allows the user to access the Application environment as soon as the database is created. Menu Bar #1 is called automatically when the **Test Application** command is chosen in the **Run** menu.

The default menu bar includes three menus: **File**, **Edit** and **Mode**.



- **File**: This menu only includes the **Quit** command. The *Quit* standard action is associated with the command, which causes the application to quit.
- **Edit** (standard): The **Edit** menu is standard and completely modifiable. Editing functions such as copy, paste, etc. are defined using standard actions.

Compatibility Note Beginning with version 11 of 4D, it is no longer possible to keep the former mechanism of the **Edit** menu (managed by the system). A standard **Edit** menu is added when databases are converted. A dialog box informs you which menu bars have been modified. Take note of the menu bars concerned because you must increment the value of the *menu* parameter in any code that manages and executes these menu bars. In fact, the **Edit** menu was not counted in the previous functioning.

- **Mode**: The **Mode** menu contains, by default, the **Return to Design mode** command, which is used to exit the Application mode.

Note Menu items appear in *italics* because they consist of references and not hard-coded text. For more information about this point, please refer to the "Using a Reference for a Label" paragraph on page 874.

You can modify this menu bar as desired or create additional ones.

Creating a Menu Bar

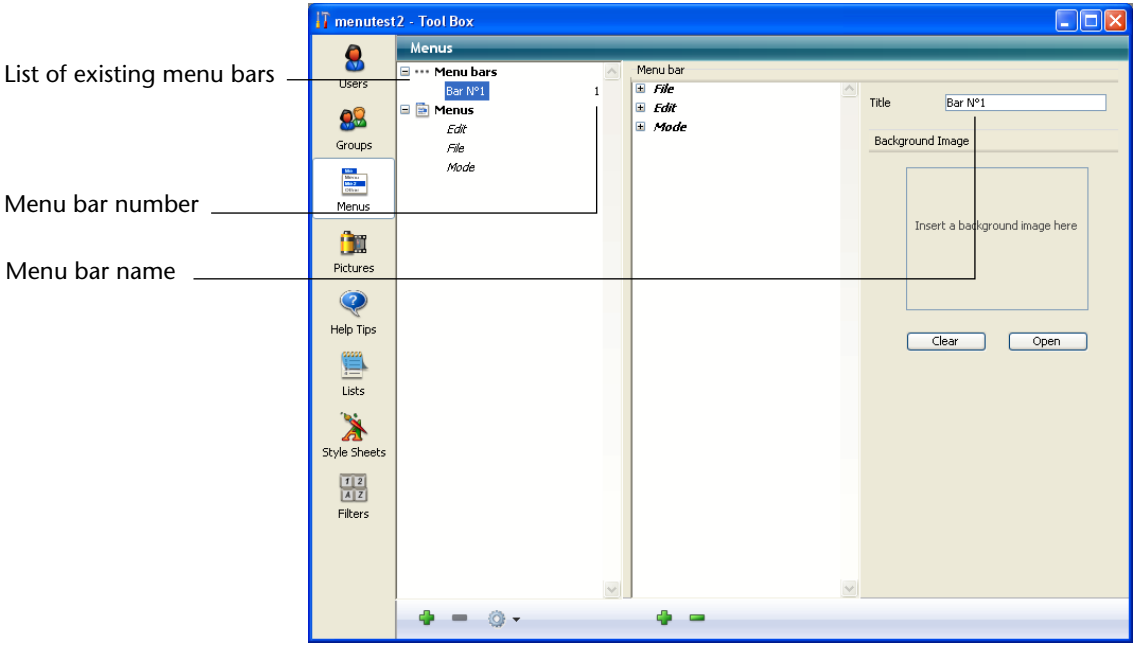
This section describes the process of creating a custom menu bar.

Note It is possible to create menu bars entirely by programming. For more information about this point, please refer to the *4D Language Reference* manual.

► To create a menu bar:

1 Display the "Menus" page of the 4D tool box.

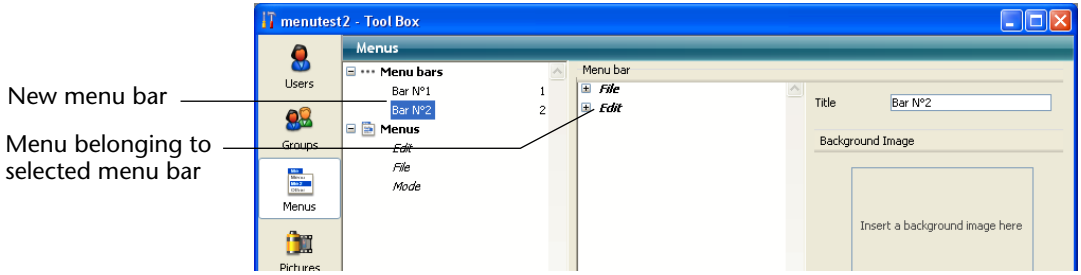
4D displays the Menu Bar editor. By default, Menu Bar #1 appears in the panel on the left — as well as any other menu bars that may already have been created.



4D assigns menu bar numbers sequentially — Menu Bar #1 appears first. You can rename menu bars but you cannot change their numbers. These numbers are used by the language commands.

2 Click the add button ✚ below the menu bar area.
OR
Choose Create a new menu bar from the contextual menu of the list or the options menu below the list.

A new menu bar appears in the list containing the default menus (**File** and **Edit**).



3 Alt+click (Windows) or Option+click (Mac OS) on the name of the menu bar or click twice on it in order to switch it to editing mode and enter a custom name (optional).
OR
Enter the custom name in the "Title" area.

A custom name can facilitate the identification of menu bars in the different dialog boxes of the Design environment and in the language commands. A menu bar name may contain up to 31 characters and must be unique.

At this point, you can begin modifying menu bars or adding menus, commands to the menus, etc.

Creating Menus

You can create menus at any time. They can then be attached to menu bars or to other menus (hierarchical submenus), or they can not be attached to any menu bar (or menu) and be handled by programming only (independent menus).

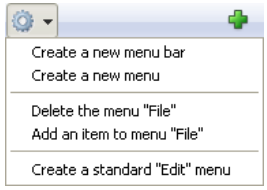
You don't have to create the menus in the order that they will eventually appear. You can rearrange menus after you've created them using drag and drop. For more information, see the section "Rearranging Menus and Menu Commands" on page 874.

Note It is possible to create menus entirely by programming. For more information about this point, please refer to the *Language Reference* manual.

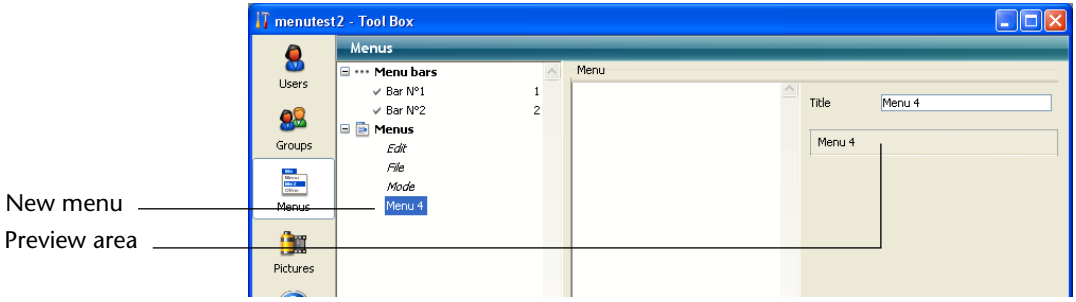
► To create a menu:

1 **Display the "Menus" page of the 4D tool box.**

2 **Select the "Menus" title or an existing menu in the list of source menus and click on the button.**
OR
Choose Create a new menu in the contextual menu (click on the "Menus" title or on an existing menu) or in the options menu of the editor.



4D adds a new menu to the bottom of the list.



3 **Alt+clicK (Windows) or Option+click (Mac OS) on the menu name or click on it in order to switch it to editing mode and enter a custom name.**
OR
Enter the custom name in the "Title" area.

You can enter the menu name as "hard coded" or enter a reference for a variable, a resource or an XLIFF element. For more information about this point, please refer to the paragraph "Using a Reference for a Label" on page 874.
If you enter the name directly, you must avoid any control characters that may disturb the menu display (see the paragraph "Using Control Characters in Menu Labels" on page 870).

4 **Repeat steps 2 and 3 to add more menus (optional).**

You can preview a menu that you are creating at any time by selecting the source menu then clicking in the preview area on the right-hand side of the editor.

Adding Menu Commands

For each of the menus, you must create the commands that appear when the menu drops down. You can insert items that will be associated with methods or standard actions, or attach other menus (submenus). Attaching hierarchical submenus is described in the paragraph “Attaching a Menu to Another Menu (Submenus)” on page 872.

► To add a menu command:

1 **In the list of source menus, select the menu to which you want to add a command.**

If the menu already has commands, they will be displayed in the central list. If you want to insert the new command, select the command that you want it to appear above. It will still be possible to reorder the menu subsequently.

2 **Choose Add an item to the menu “*MenuName*” in the options menu of the editor or from the contextual menu (right click in the central list).**
OR
Click on the add button  **located below the central list.**

4D adds a new item with the default name “*Item X*” where *X* is the number of items already created.

3 **Alt+click (Windows) or Option+click (Mac OS) or click twice on the name of the command in order to switch it to editing mode and enter a custom name.**
OR
Enter the custom name in the “Title” area.

You can enter the menu name as “hard coded” or enter a reference for a variable, a resource or an XLIFF element. For more information about this point, please refer to the paragraph “Using a Reference for a Label” on page 874.
If you enter the name directly, you must avoid any control characters that may disturb the menu display (see the following paragraph).

4 **(Optional) Enter a custom reference in the “Reference” area.**

This reference can be used by language commands. For more information about this point, please refer to the “Custom Reference” paragraph on page 879.

5 **Repeat steps 1 to 4 to add more commands.**

Using Control Characters in Menu Labels

You can set the properties of the menu commands by using control characters (metacharacters) directly in the menu command labels. For instance, you can assign the keyboard shortcut **Ctrl+G** (Windows) or **Command+G** (Mac OS) for a menu command by placing the "/G" characters in the label of the menu item label.

Control characters do not appear in the menu command labels. You should therefore avoid using them so as not to have any undesirable effects .

The control characters are the following:

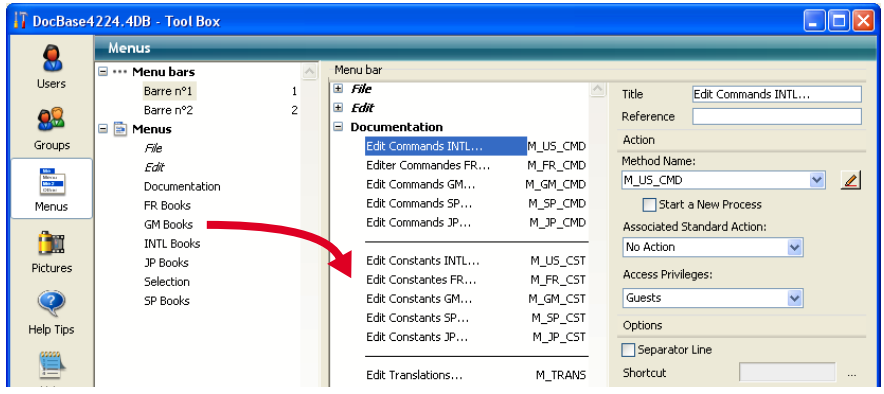
- "("
- "<"
- "!"
- "^"
- "/"

For more information on the use of these characters, refer to the description of the APPEND MENU ITEM command in the *4D Language Reference* manual.

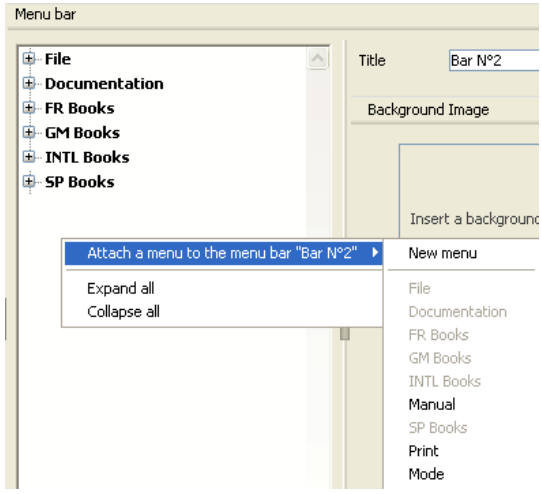
Attaching a Menu to a Menu Bar

Once you have created a menu, you can attach it to a menu bar. This can be done by drag and drop, by the options menu or by the contextual menu of the central area.

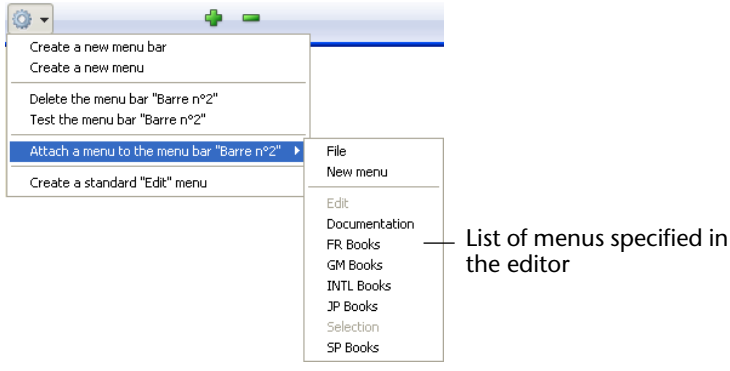
- **Using drag and drop**: Click on a menu bar in order to display its contents in the central list; select a menu in the left-hand list and drag it to the desired location in the central list:



- **Using the contextual menu**: Click on a menu bar in order to display its contents in the central list; click with the right-button in this area and select the **Attach a menu to the menu bar "bar name"** > command, then choose the menu to be used as a sub-menu:

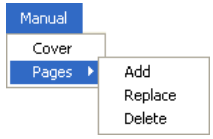


- **Using the options menu**: Select a menu bar in the left-hand list then click on the options button found below the list; select the **Attach a menu to the menu bar "bar name"** > command, then choose a menu to be used as a sub-menu:



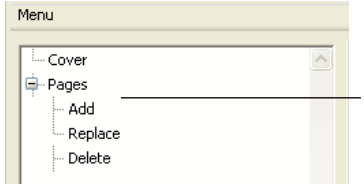
Attaching a Menu to Another Menu (Submenus)

It is possible to attach menus to other menus, in other words, to set up hierarchical submenus. In a menu bar, sub-menus can be used to group together functions organized according to subject within the same menu:



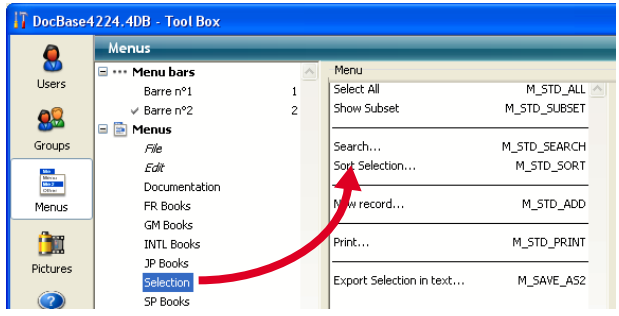
Sub-menus and their items can have the same attributes as the menus themselves (actions, methods, shortcuts, icons, and so on.)

In the Menu editor, sub-menus appear as items of a hierarchical list:

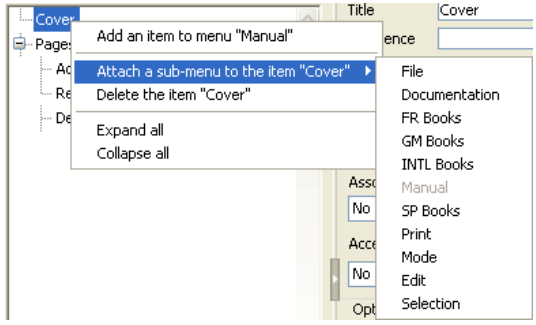


► To create a sub-menu, simply associate (attach) an existing menu with the item of another menu. There are two ways to do this:

■ **Using drag and drop**: Select a menu in the left-hand list and drag it onto the item in the central list to which you wish to attach the sub-menu:



- **Using the contextual menu**: In the central list, click with the right button on the item to which you want to attach the menu. In the contextual menu, select the **Attach a sub-menu to the item “item name”>** command, then choose the menu you want to use as sub-menu:



The menu being attached thus becomes a sub-menu. The title of the item is kept (the original sub-menu name is ignored), but this title can be modified.

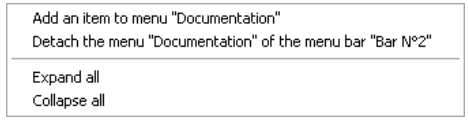
The items of the sub-menu keep their original characteristics and properties and the functioning of the sub-menu is identical to that of a standard menu.

You can create sub-menus of sub-menus. Simply expand the sub-menu in the central list and attach a menu to one of the sub-items. You can add sub-menus to a virtually unlimited depth. Note, however, that for reasons concerning interface ergonomics, it is generally not recommended to go beyond two levels of sub-menus.

Detaching a Menu or Sub-menu

You can detach a menu from a menu bar or a sub-menu from a menu at any time. The detached menu is then no longer available in the menu bar or sub-menu as the case may be, but it is still present in the list of menus.

To detach a menu, click with the right button on the menu or sub-menu that you want to detach in the central list, then choose the **Detach the menu “menu name” of the menu bar “bar name”** or **Detach the sub-menu of the item “item name”**:



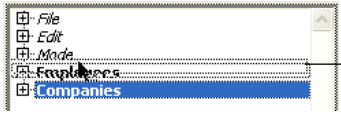
Independent Menus

It is possible to set “independent” menus; in other words, menus that are not attached to a menu bar or to another specific menu. These menus can be configured in the Menu editor but must be managed using language commands.
To create an independent menu, select the **Menu** command from the menu associated with the creation button under the menu bar/menu list:

Rearranging Menus and Menu Commands

After you create the menus for a menu bar and the menu commands for a menu, you can reorder them using drag and drop. To insert a menu command at a different place in the order, simply drag it to the new location. To move a menu, simply drag it to another location in the list of menus.

The following illustration shows a menu being dragged:



Companies menu being dragged to a new position

Using a Reference for a Label

It is possible to enter menu labels and titles as references and not as “hard-coded” text. This will facilitate the translation of applications. You can use three types of references:

- An XLIFF resource reference of the type *:xliff:MyLabel* or *:10115,3* (compatibility). For more information about XLIFF references, please refer to the annexe “XLIFF Architecture”, page 1293.
- An STR# resource reference. If you enter, for example, *:2000,3* the character string contained in the third line of the STR# 2000 resource will be displayed as the menu label. Changing the contents of this STR# resource by programming using the 4D language or a resource editor (under Mac OS) will change the name of the menu the next time it is displayed.
- An interprocess variable name followed by a number, for example *<>vlang,3*. Changing the contents of this variable will modify the menu label when it is displayed. In this case, the label will call an XLIFF or STR# resource depending on the translation architecture of the database:

- *XLIFF Architecture*: The value contained in the <>*vlang* variable corresponds to the id attribute of the group element. The second value (3 in this example) designates the id attribute of the trans-unit element.
- *STR# Architecture*: The character strings contained in the third line of the STR# resource whose number is found in the <>*vlang* interprocess variable will be displayed as the menu label.

Deleting Menus and Menu Items

You can delete a menu bar, a menu or a menu item at any time. Note that each menu or menu bar has only one reference. When a menu is attached to different bars or different menus (as a submenu), any modification or deletion made to the menu is immediately carried out in all other occurrences of this menu.

To delete a menu bar, menu or menu item, you have two possibilities:

- Select the item to be deleted and click on the deletion button  located beneath the list.
- Use the **Delete the menu bar “”**, **Delete the menu “”** or **Delete the item “”** commands from the contextual menu or the options menu of the editor.

Note It is not possible to delete Menu Bar #1.

Specifying the Action of a Menu

To enable a menu command to perform its function, you must assign either a **project method** or a **standard action** to it.

These methods or standard actions perform the functions indicated by the menu commands. For example, the **Monthly Report** menu command can call a project method that prepares a monthly report from a table containing financial data. The **Cut** menu command can call the *Cut* standard action in order to move the selection to the clipboard and erase it from the window in the foreground. When a menu command is chosen, 4D executes the standard action or project method that is assigned to it.

The choice between associating a standard action or a project method with a menu command depends on the type of result desired. In principle, it is preferable to choose a standard action whenever possible since they implement optimized mechanisms, more particularly an activation/deactivation according to the context.

Note Standard actions are not compatible with the mechanism that converts menu bars into lists of URLs, which is provided by the Web server of 4D in contextual mode. For more information on this point, refer to the *4D Language Reference* manual.

You can also assign both a standard action and a project method to a menu item. In this case, the standard action is never executed; however, 4D uses this action to enable/disable the menu item according to the context. When a menu item is disabled, the associated project method cannot be executed.

You create the project methods in the Method editor. You can create them either before or after you assign them to the menu command. When you have assigned a method to a menu command in the Menu Bar editor, you can open this method by simply selecting the  button.

If a menu command is not associated with a standard action or a method, when it is selected in Application mode, 4D will automatically return to the Design mode (if it is accessible).

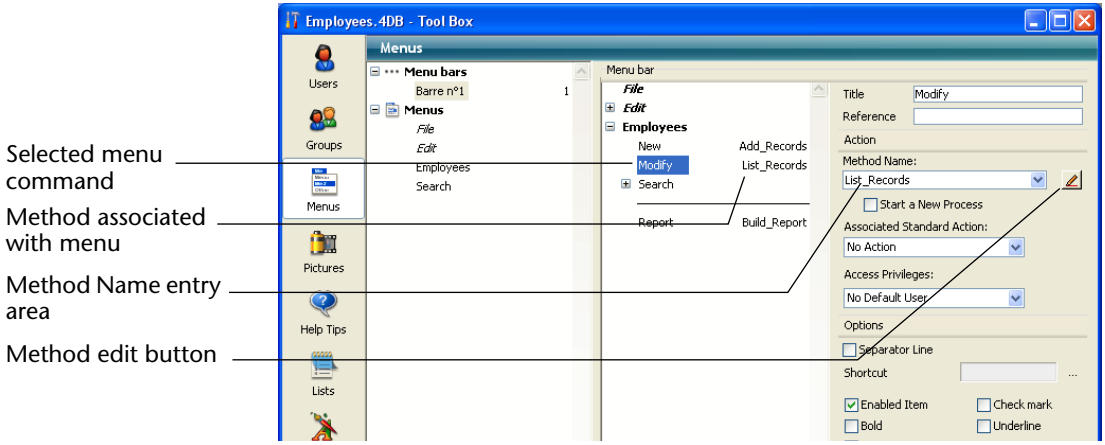
Associating a Project Method

► To assign a project method to a menu command:

1 **Create or select the menu command.**
The properties area changes to display the properties of the selected menu command.

2 **If the project method already exists in the database, select it using the "Method Name" combo box.**
OR
If the project method does not exist, enter its name in the "Method Name" combo box then click on the **button.**

In the latter case, 4D displays the project method creation dialog that is used to access the Method editor.



Note If you change the name of a method that is used in a menu, you must update the method name here in the Menu Bar editor.

3 Click the Start a New Process check box (optional).

If you click the **Start a New Process** check box, a new process is created when the menu command is chosen.

Start a New Process

Normally, a method attached to a menu command executes within the current process unless you explicitly call New process in your code. The **Start a New Process** check box makes it easier to start a new process.

If you click the **Start a New Process** check box, 4D will create a new process when the menu command is chosen. In the Process list, 4D assigns the new process a default name using the format *M_ProcessNumber*. The names of processes started from a menu are created by combining the prefix "M_" with the process number.

For more information about processes, see the chapter "Processes" on page 929.

Note When the menu command is called via a URL from the Web server home page in contextual mode, the server receives the special 4DMETHOD URL if the **Start a New Process** option is checked. In this case, you must be sure that the method has the "Available through 4DACTION, 4DMETHOD and 4DSCRIPT" attribute (for more information, refer to the "Defining the Properties of Project Methods" paragraph on page 773.

Associating a Standard Action

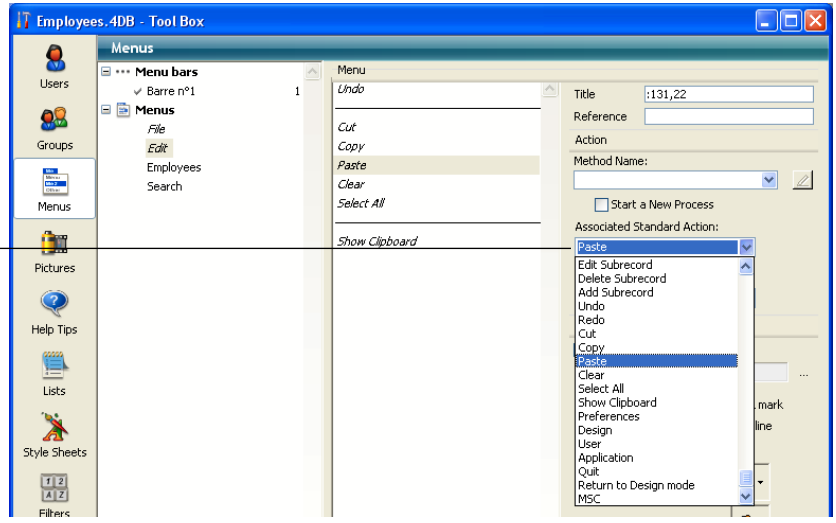
► To associate a standard action with a menu command:

1 **Create or select the menu command.**

The properties area changes to display the properties of the selected menu command.

2 **Choose the action you want to assign to it in the "Associated Standard Action" menu.**

Selection of a standard action for the selected menu command



The list of standard actions proposed for menus is similar to the one for buttons (accessible in the Property List for buttons in the 4D Form editor). Most of the actions can, in fact, be used in both contexts. Only the *Automatic splitter* action cannot be associated with a menu command: therefore it does not appear in the pop-up selection menu. For a description of each standard action, refer to the "Standard Button Actions" paragraph on page 555.

Note for Mac OS Under Mac OS, the custom menu commands associated with the **Preferences** and **Quit** actions are automatically placed in the application menu, in compliance with the platform interface standards.

Setting Menu Properties

You can associate custom references to menu commands, change the font style, add separator lines between groups of menu commands, assign a keyboard shortcut for a menu command, and enable or disable menu commands. You can also associate an icon with a menu command and create a custom toolbar.

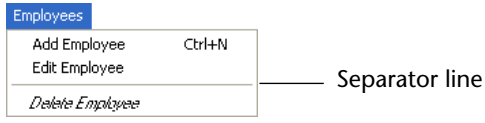
Custom Reference

It is possible to associate a custom reference with each menu item. A menu item reference is a character string whose contents can be freely chosen.

Menu item references are mainly useful for programmed management of menus, in particular when using the Dynamic pop up menu command.

Separator Lines

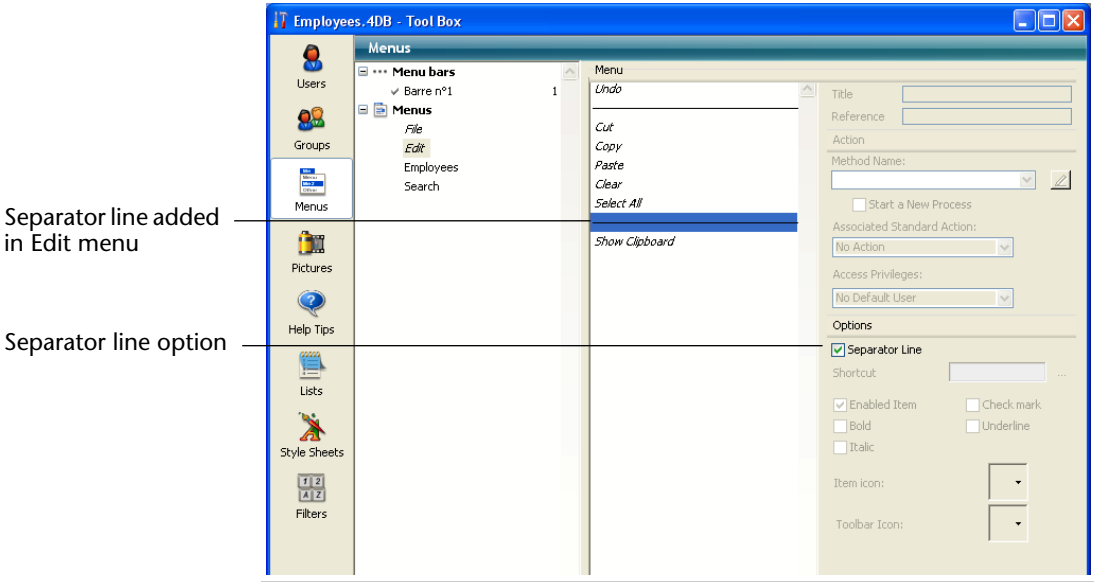
Groups of menu commands in a menu can be divided by a separator line. This convention is useful for grouping associated menu commands by function.



You add a separator line by creating a menu command. Instead of entering the menu command's text in the current menu bar area, you simply select the **Separator Line** option. Instead of text, a line appears in the current menu bar area.

Note Under Mac OS, if you use the dash "-" as the first character of a menu item, it will appear as a separator line. This is especially helpful when using the APPEND MENU ITEM command.

The line appears in the current menu bar area.



Note When the **Separator Line** option is checked, the other properties have no effect.

Assigning Keyboard Shortcuts

You can add keyboard shortcuts to any menu command. If a menu command has one of these keyboard shortcuts, users will see a symbol for it next to the menu command. For example, "**Ctrl+C**" (Windows) or "⌘**+C**" (Mac OS) appears next to the **Copy** menu command in the **Edit** menu. This means you can copy a selection by holding down the **Ctrl** key and pressing **C** under Windows, or holding down **Command** and pressing **C** under Mac OS.

You can also add the **Shift** key as well as the **Alt** (Windows) or **Option** (Mac OS) keys to the shortcut associated with a menu command. This multiplies the number of shortcuts that can be used with the menu bars created. The following types of keyboard shortcuts can therefore be defined:

- Under Windows:
  - **Ctrl+*letter***
  - **Ctrl+Shift+*letter***
  - **Ctrl+Alt+*letter***
  - **Ctrl+Shift+Alt+*letter***

- Under Mac OS:
  - **Command+*letter***
  - **Command+Shift+*letter***
  - **Command+Option+*letter***
  - **Command+Shift+Option+*letter***

Note. It is advisable to keep the default keyboard shortcuts associated with standard actions.

You can use any alphanumeric keys as a keyboard shortcut, except for the keys reserved by standard menu commands that appear in the **Edit** and **File** menus, and the keys reserved for 4D menu commands.

These reserved key combinations are listed in the following table.

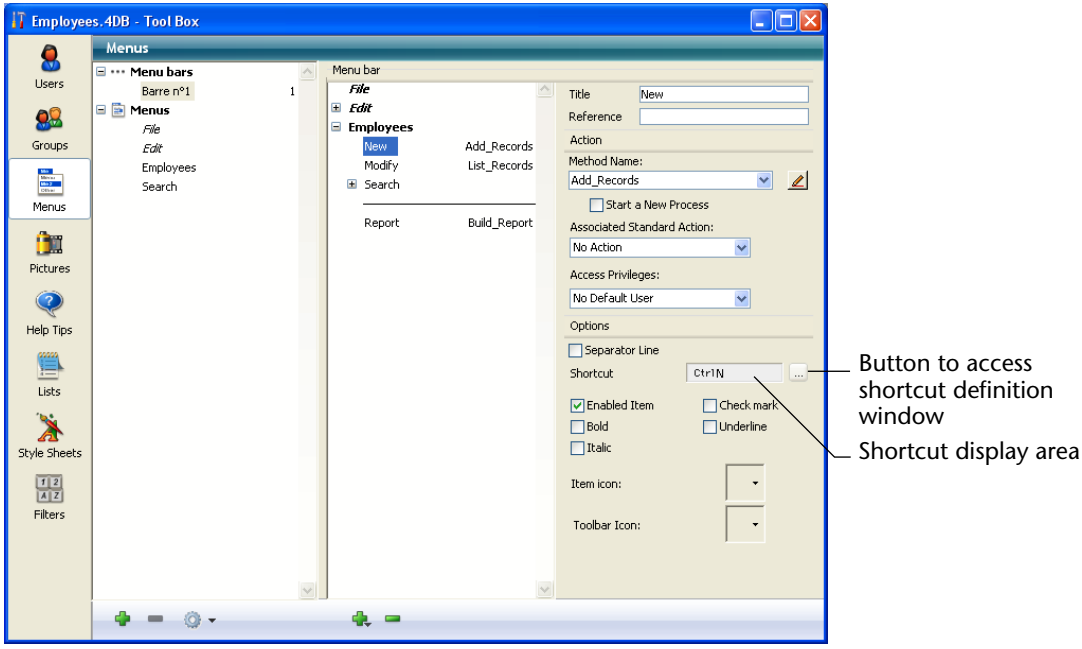
Key[1]	**Operation**
Ctrl+C	Copy
Ctrl+Q	Quit
Ctrl+V	Paste
Ctrl+X	Cut
Ctrl+Z	Undo
Ctrl+. (period)	Stop action

1. Under Mac OS, use Command instead of Ctrl.

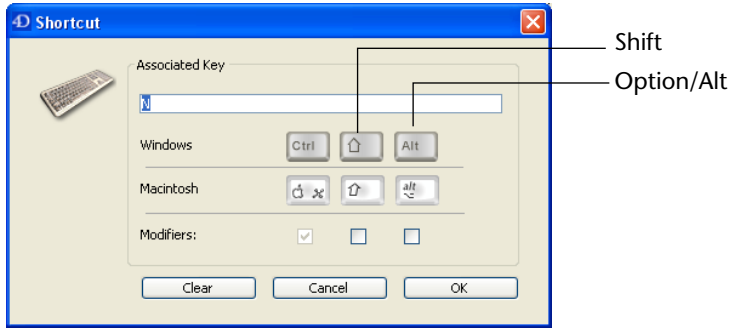
▶ To assign a keyboard shortcut:

1 Select the menu item to which you want to assign a keyboard shortcut.

2 Click on the ... button to the right of the "Shortcut" entry area.



The following window appears:



3 Enter the character to use then (optional) check the Shift and/or Alt (Option) options according to the combination desired.

Notes

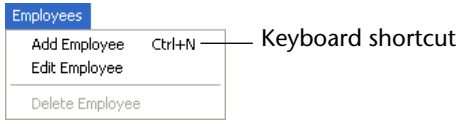
- You can also directly press the keys that make up the desired combination (do not press the Ctrl/Command key).
- It is not possible to deselect the Ctrl/Command key, which is mandatory for keyboard shortcuts for menus.

To start over, click on **Clear**.

4 Click OK to validate the changes.

The shortcut defined is shown in the “Shortcut” entry area:

The keyboard shortcut appears in the menu as follows.



Note An active object can also have a keyboard shortcut. If the **Ctrl/Command** key assignments conflict, the active object takes precedence. For information on assigning keyboard shortcuts to active objects, see the section “Assigning a Keyboard Shortcut” on page 542.

Active Item

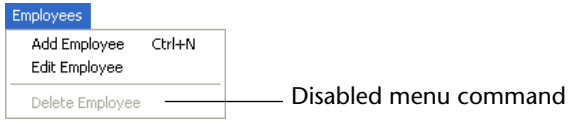
You can specify whether a menu item will appear enabled or disabled. An enabled menu command can be chosen by the user; a disabled menu command is dimmed and cannot be chosen.

Unless you specify otherwise, 4D automatically enables each menu command you add to a custom menu. You can disable an item in order, for example, to enable it only using programming (ENABLE MENU ITEM and DISABLE MENU ITEM commands).

Note You can enable or disable menu commands using methods.

When the **Enabled Item** check box is unchecked, the menu command appears dimmed, indicating that it cannot be chosen.

A disabled command appears as follows:



Check Mark

This option can be used to associate a system check mark (√) with a menu item. You can then manage the display of the check mark using language commands (SET MENU ITEM MARK and Get menu item mark).

Check marks are generally used for continuous action menu items and indicate that the action is currently underway:

Font Styles

4D lets you customize menus by applying different font styles to the menu commands. You can customize your menus with the Bold, Italic or Underline styles.

Be cautious when applying font styles to your menus — too many styles will be distracting to the user and give a cluttered look to your application.

To apply a style, select the menu command you want to modify and then choose the style using the check boxes.

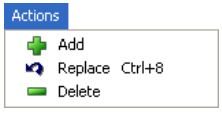
Item Icon

The **Item icon** option can be used to associate an icon with the selected item. The icon to be used must be stored beforehand in the 4D picture library. When you click on the area associated with this option, a hierarchical pop-up menu appears so that you can select a library picture:

Note If the picture library is empty, the **Library Pictures** item is not displayed.

Once set, the item icon appears in the preview area:

It is displayed directly in the menu, next to the item:



To remove the icon from the item, choose the **No Icon** option from the "Item Icon" area menu.

Toolbar Icons

You can associate a tool bar icon with a menu item. When you do so, the icon is used as a button in the toolbar associated with the menu bar in Application mode.

The icons to be used must have been stored beforehand in the 4D picture library. Selecting the picture is carried out in the same way as for menu item icons (see previous paragraph).

The icon will be displayed in a 20 x 20 pixel frame. 4D automatically uses the text of the menu command as the help tip of the toolbar button.

Note It is possible to hide the Application environment toolbar in the Preferences of the application (see the "Display Toolbar" paragraph on page 165).

To delete a tool bar icon associated with a menu command, choose **No icon** in the "Toolbar Icon" drop-down list.

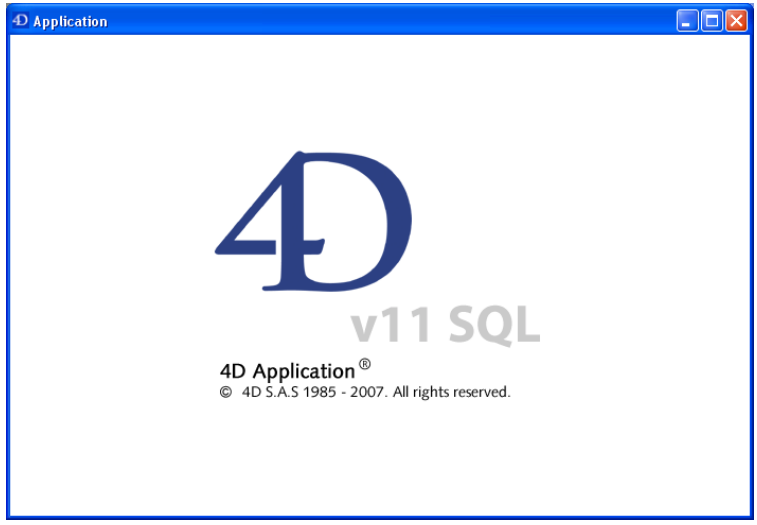
Overall Management of Menu Bars

4D lets you associate a custom splash screen picture with each menu bar and to preview this menu bar at any time.

Setting a Splash Screen

You can enhance the appearance of each menu bar by associating a custom splash screen with it. The window containing the splash screen is displayed below the menu bar when it appears. It can contain a logo or any type of picture.

By default, 4D displays the 4D logo in the splash screen:



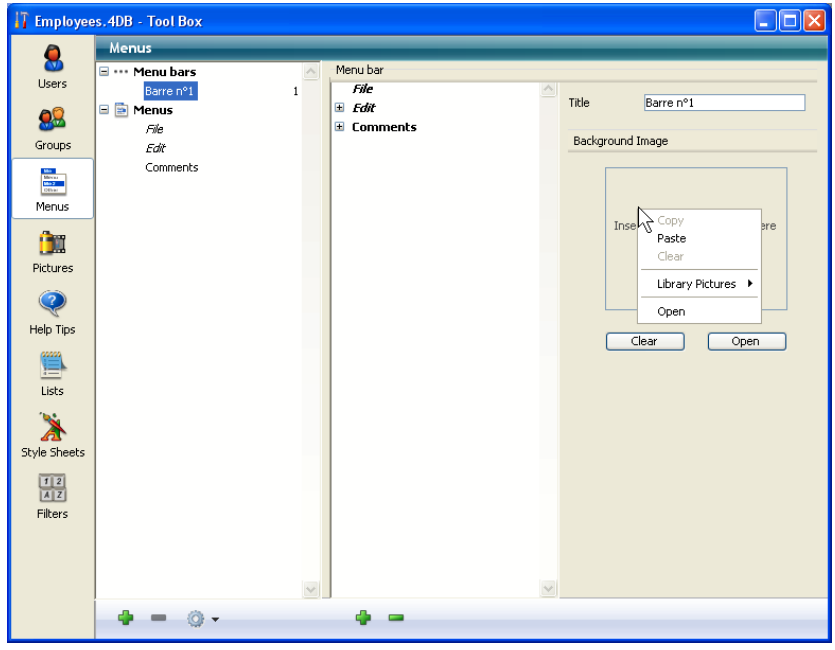
A custom splash screen picture can come from any graphic application. 4D lets you paste a clipboard picture, use a picture from the library or use any picture present on your hard disk. Any standard picture format supported by 4D can be used.

► To modify the splash screen picture:

1 **Select the menu bar with which you want to associate the custom splash screen.**

2 **To open a picture stored on your disk directly, click on the Open button.**
OR
Click in the "Background Image" area.

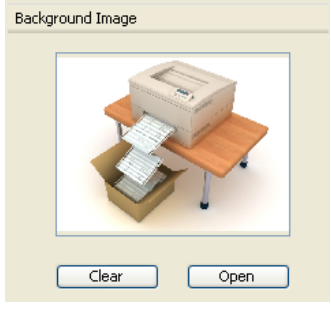
A pop-up menu appears, which provides various options for adding a picture.



- To paste a picture from the clipboard, choose **Paste**.
- To select a picture stored in the picture library, choose **Library Pictures** (if the picture library is empty, this item is not displayed).
- To open a picture stored in a disk file, choose **Open**.

3 **If you choose Open, a standard Open file dialog box will appear so that you can select the picture file to be used.**

Once set, the picture is displayed in miniature in the area. It is then associated with the menu bar.



You can view the final result by testing the menu bar (see the following paragraph).

In Application mode, the picture is displayed in the splash screen with the “Truncated (Centered)” type format.

Note You can choose whether to display or hide this window in the database Preferences (see the “Display Windows” paragraph on page 165).

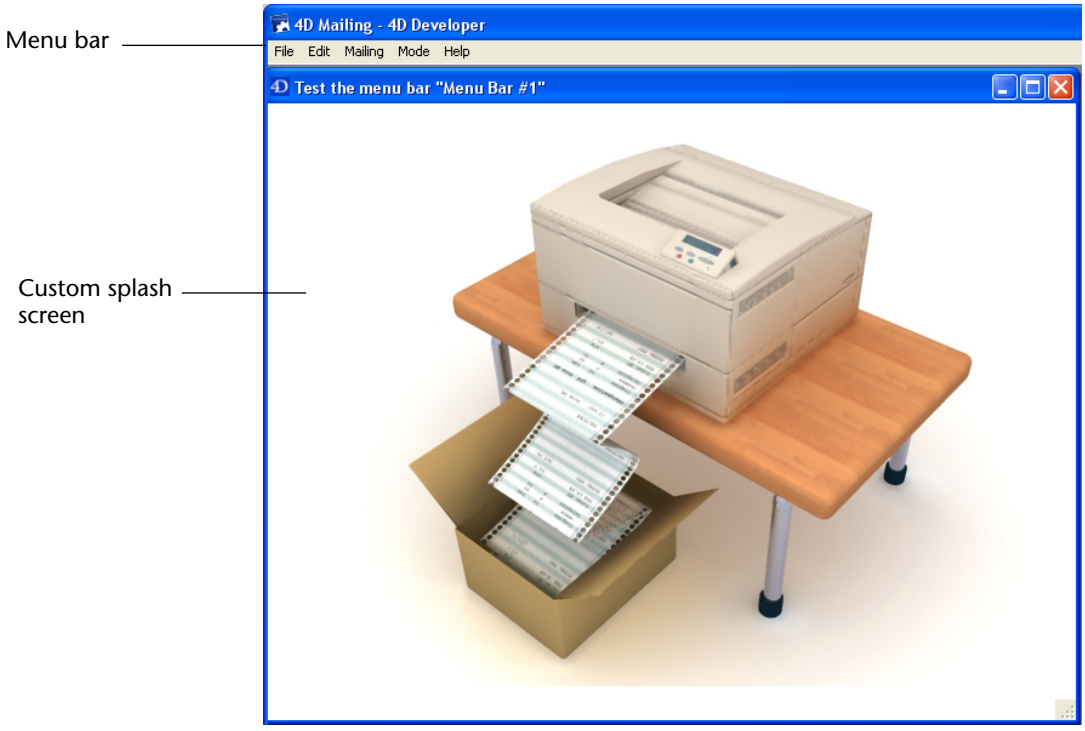
To remove the custom picture and display the default one instead, click on the Clear button or click in the “Background Image” area and choose **Clear** in the pop-up menu.

Previewing Menu Bars

The Menu Bar editor lets you view the custom menus and splash screen at any time, without closing the tool box window.

To do so, simply select the menu bar and choose **Test the menu bar “Menu Bar #*X*”** in the contextual menu or the options menu of the editor.

4D displays a preview of the menu bar as well as the splash screen.



Note that this preview does not let you view any toolbar that may have been associated with the custom menus.
You can scroll down the menus and submenus to preview their contents. However, these menus are not active.
To test the functioning of menus and the toolbar, you must use the **Test Application** command from the **Run** menu.

Menus and Custom Applications

Menu bars provide the major interface for custom applications. For each custom application, you must create at least one menu bar with at least one menu. See the *4D Language Reference* manual for more information about creating custom applications.

You can create menu bars for the Application environment regardless of whether you are creating a custom application or simply creating menus for use in the Application environment. By default, Menu Bar #1 is the menu bar displayed in the Application environment. You can change which menu bar is displayed using a method.

If you define a menu command without assigning it a method, choosing that menu command exits the Application environment and returns to the Design mode (if access to this mode is allowed). If you are using the application with 4D Desktop, leaving the Application environment causes you to exit the application.

If you are using the full 4D application, a password access system can be set up to control where each user is placed after leaving the Application environment. In fact, you can define an access group for the Design Environment in the Preferences dialog box.

Users who do not belong to the access group set for the Design environment will not be able to access it from the Application mode by either selecting a menu command, or by pressing the **Alt+Shift+right click** (Windows) or **Control+Option+Command+click** (Mac OS) shortcut, which displays the process pop-up menu. When users that do not have the adequate access privileges attempt to switch to the Design environment, 4D quits.

Note The Designer and the Administrator will always have access to the Design environment, even if they do not belong to the Design environment access group. For more information, refer to the chapter "Users and Groups" on page 831.

14 Picture Library

Use the Picture Library to store graphics that you can use as design elements in forms, as toolbar or list icons, picture menu items, or picture buttons. With the Picture Library, you can use a graphic in several places in your database but you only need to store it in one place. When you update a picture in the Picture Library, all references to the picture are updated automatically. This feature can reduce the size of your Structure files and make changes to the database easier to manage.

4D supports the most current picture formats, in particular the JPEG, SVG, PNG, BMP, GIF and TIFF formats (non-exhaustive list). In the picture library, the pictures will be stored in their original format, without any interpretation. The specific features of the different formats (shading, transparent areas, etc.) will be retained and displayed without alteration. The PICTURE CODEC LIST command can be used to get the native types found on the machine.

The Picture Library includes a 2D Paint editor that allows you to create or retouch pictures. It is an ideal environment to design buttons and icons — in this case, the pictures are stored in the Pict format.

In addition to the features mentioned above, the Picture Library also includes integrated functions that can create or edit tables of thumbnails in order to create picture buttons or picture menus.

Overview

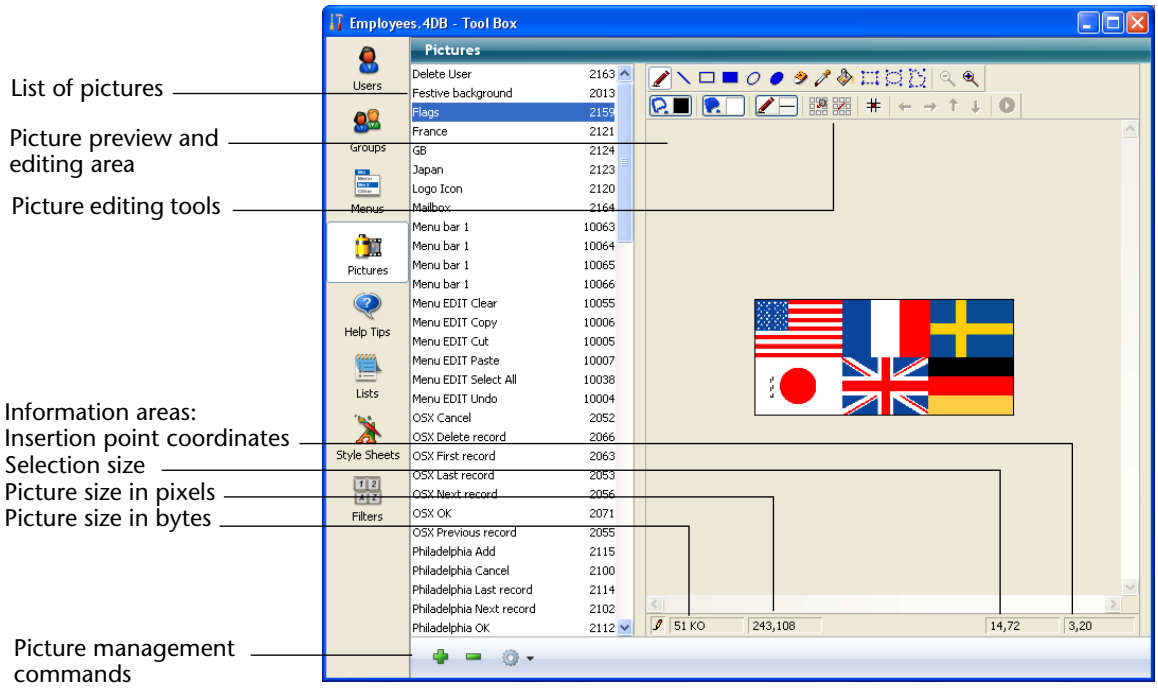
The Picture Library is included in the tool box of 4D.

► To display the Picture Library:

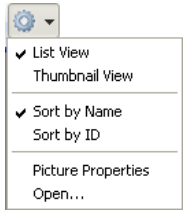
1 Choose Tool Box > Picture Library from the Design menu.
OR

Click on the "Tool Box" button of the 4D tool bar then click on the "Pictures" button.

The picture library window displays the list of pictures stored in the database. It includes picture management commands as well as a tool-bar for editing picture contents:

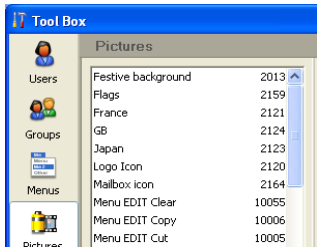


You can set the display mode and sort criterion of the picture library list using commands associated with the options button located below the list:



- Choice of display mode (list or thumbnail)

Display as **List** (default mode)



Display as **Thumbnail**

- You can also choose to sort pictures by name (default) or by ID number. The sort is always ascending.
  If you choose the **Sort by ID** command and the pictures are displayed as thumbnails, the ID numbers will appear under each thumbnail instead of the names.

Adding Pictures to the Library

You can add pictures to the Picture Library in three ways:

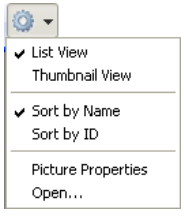
- By importing a picture file
- By pasting a picture from the Clipboard (or using drag and drop).
- By creating a new picture (blank) in which you draw its contents.

Importing a Picture File

4D can import and display the most current picture formats, such as the JPEG, SVG, PNG, BMP, GIF and TIFF formats (non-exhaustive list). If you import a picture whose format is not supported natively by 4D, the program will call on QuickTime® routines to interpret it (QuickTime must be installed on your machine).

► To import a new picture into the Picture Library:

1 **Choose Open... in the options menu of the library:**



A standard Open file dialog box appears. The “Files of type” menu lets you display the file formats supported by 4D.

2 **Select the file to be imported and click Open.**

The Picture Properties dialog is displayed. The photo's dimensions are automatically defined depending on the picture imported.

For more information on picture properties, refer to "Setting Picture Properties" on page 895.

3 **If necessary, modify the name and ID number as well as other properties and click OK to create the picture in the Picture Library.**

Note You can only modify the ID number of the picture when it is being created in the Picture Library.

Pasting or Dropping an External Picture

You can create a picture in the library by pasting a picture from the clipboard or dropping a picture file from another application (another 4D database, a system window, Web browser, and so on).

► To add an external picture by copy-paste or drag and drop:

1 **Copy a picture to the Clipboard.**
OR
Drag an external picture file and drop it onto the Picture Library list area.

2 **(Copy-paste only) Select the Picture Library list area and press the Ctrl+V (Windows) or Command+V (Mac OS) keys or choose Paste from the Edit menu of 4D.**

The Picture Properties dialog is displayed. The dimensions that appear in the dialog are the picture's dimensions and its default name is *From ClipBoard*.

For more information on picture properties, refer to "Setting Picture Properties" on page 895.

3 **If necessary, modify the ID number as well as other properties and click OK to create the picture in the Picture Library.**

Note You can only modify the ID number of the picture when it is being created in the Picture Library.

Creating a New Picture

The Picture Library contains an integrated graphics editor. You can create a blank picture and draw its contents using the editor.

► To create a new picture:

1 **Click on the add button located below the list of pictures.**

The Picture Properties dialog box is displayed. By default, the dimensions of the new pictures are 100x100 pixels.

For more information on picture properties, refer to "Setting Picture Properties" on page 895.

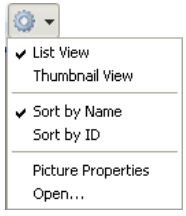
2 If necessary, modify the name and ID number as well as other properties and click OK to create the picture in the Picture Library.

Note You can only modify the ID number of the picture when it is being created in the Picture Library.

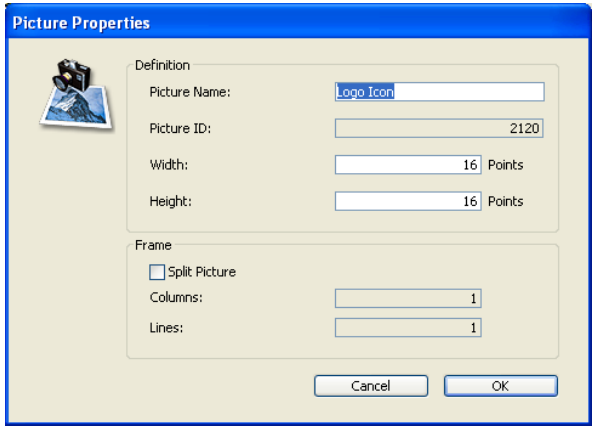
Setting Picture Properties

The Picture Properties dialog box allows you to set or display the picture's name, dimensions, and frame attributes. It also displays the ID number of the picture (not modifiable).

No matter how you add a picture, the dialog box shown below appears.



You can display this dialog box at any time by **double-clicking** a picture or by selecting a picture and choosing **Picture Properties** from the options menu of the library.



This dialog box displays the following properties:

- **Picture Name**: You can assign the same name to several pictures, only the ID number has to be different.
- **Picture ID**: Unique ID number of the picture. This number is the reference number for the picture. It is how you refer to the picture when creating picture buttons, picture pop-up menus, custom toolbars, lists or when you handle pictures programmatically.

Note You can set this number when you create the picture, but you cannot modify it afterwards.

- **Width** and **Height**: Size of the picture. These values are precalculated when you import a picture (from a file or from the Clipboard). When you split the picture (see below), the values correspond to the size of each frame.
- **Frame area**: Allows you to create thumbnails from a single picture for use in creating an array of buttons or picture menus. For more information, please refer to the "Creating Thumbnails" paragraph on page 898.

To create the picture, validate the dialog box. Once the picture's properties have been set, you can modify them at any time by selecting the picture and choosing **Picture Properties...** from the library's **Picture** menu.

Modifying Picture Contents

The Picture Library has a toolbar (like a Paint program) that allows you to draw and modify pictures.

To create or modify a picture, you just have to click on the view/edit area.

IMPORTANT: *The Picture Library's editor works only in bitmap mode. If you modify a vector graphic in the editor, it will be transformed into a bitmap and its original characteristics will be lost when the modified picture is saved. Moreover, the editor does not support the alpha channel (transparency information). If you modify a picture originally containing an alpha channel, this information will be lost when the modified picture is saved.*

The Picture Library's graphic tools are the following:

Icons	Tools	Function	Option keys
	Pen	Draws point by point	Alt (Option): Allows you to take the color above which the cursor is located
	Line	Draws a line	Shift: The angles are in multiples of 45°
	Hollow Rectangle + Full Rectangle	Draws a hollow rectangle Draws a full rectangle	Shift: Draws squares Ctrl (Command): The rectangle is drawn from its center
	Hollow Oval + Full Oval	Draws a hollow oval Draws a full oval	Shift: Draws circles Ctrl (Command): The oval is drawn from its center
	Eraser	Erases by using the current background color	
	Color Picker	Modifies the line color by using a color from the picture	
	Flood Fill	Fills an enclosed area with the current background color	
	Selection tools	Creates a selection	
	Zoom	Zooms the picture	
	Outline Color + Fill Color	Line color and background color menus	These options can be modified from the editor's contextual menu (click with the right mouse button)
	Line width	Line width menu	

You can also use the standard editing commands (such as copy, paste, etc.) in the **Edit** menu of 4D, or the standard keyboard shortcuts.

Note If you use the standard **Paste** command while the picture editing area is selected, the Clipboard's contents are inserted into the currently edited picture. If you use the standard **Paste** command while the picture list is selected, a new picture is added to the list.

Saving and Cancelling Modifications

Any modification made to a picture is automatically saved in the library as soon as you click outside of the picture editing area (which means as soon as the editor area "loses" the focus).

IMPORTANT: *Once the picture has been saved, you cannot undo any of the modifications that you have made to it.*

While modifying a picture, you can cancel the last modification made to it by choosing **Undo** from the **Edit** menu in 4D, as well as by using the standard shortcut **Ctrl+Z** (Windows) or **Command+Z** (Mac OS).

You can also cancel all the modifications made to a picture by choosing **Revert...** from the **File** menu of 4D.

Creating Thumbnails

The Picture Library contains a set of functions that allow you to create and modify a row by column array of pictures for use in picture buttons or picture menus. The array may be either one- or two-dimensional.

In the Picture Library, such an array is called “Frames.” Elsewhere in 4D, an array of pictures may be called “thumbnails” or a “table” of pictures. The idea consists of splitting a picture using rows and columns; each cell is therefore considered a “frame” or “thumbnail.” 4D takes care of displaying the correct frame in the picture button or picture menu according to the specified parameters (for more information refer to “Picture Buttons” on page 566 and “Picture Pop-up Menus” on page 579).

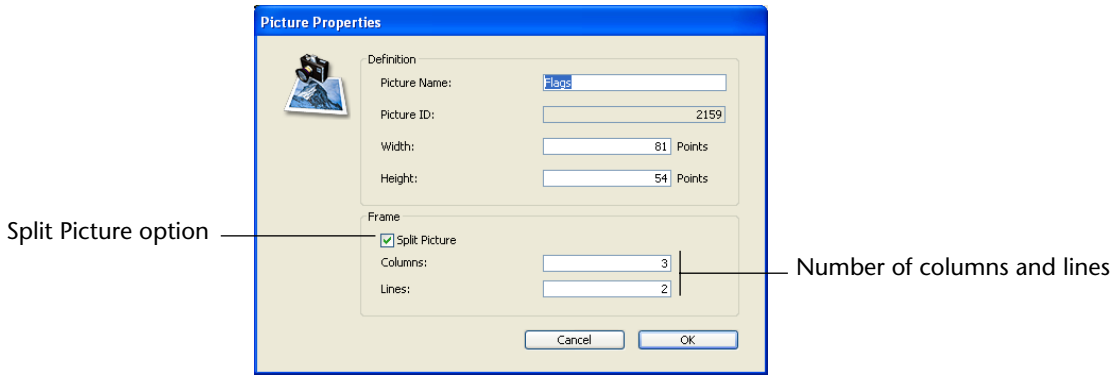
You can set a sequence of frames when creating a picture or even afterwards. You can also insert or delete lines, columns, or frames in an already defined sequence of frames.

Creating a Sequence of Frames

You can create a sequence of frames from a picture already placed in the Picture Library or when you add a picture to the Picture Library.

In both cases, you set how the picture is to be divided into separate frames in the Picture Properties dialog box. If you are creating a picture, the dialog box appears automatically. Otherwise, double-click on the picture or select it and choose **Picture Properties** from the options menu of the library.

The Frame area allows you to set the number of lines and columns of your frame sequence. To create frames, you must first check the **Split Picture** option:



Size of the Frames

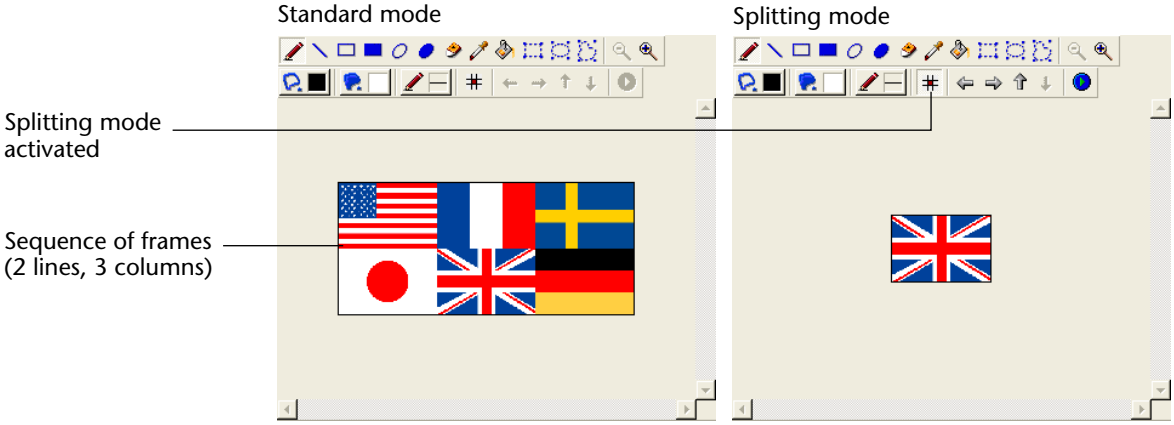
The size of the frames is automatically calculated by 4D. When you set a sequence of frames, the “Width” and “Height” areas are modified and the size of each frame is displayed.

If you want to modify the size of the frame later, you just have to enter new values into the Width and Height areas without worrying about the global size of the picture. Each resulting frame will be centered automatically (without the picture being distorted) in the new size if it is bigger. If the new size is smaller, each frame will be truncated.

Viewing the Frames

You can preview each frame to check the sequence’s appearance by using specific tools in the Picture Library’s toolbar:

When you activate the splitting mode, 4D splits the picture and displays the first frame (located in the upper left corner). You can then navigate through the frames using the arrow buttons.



You can also display all the frames automatically. This function is particularly useful if you want to create picture buttons that display in a continuous sequence. To do so, click on the **Test Animation** button . The frames appear in a continuous manner. To stop the test, click the **Test Animation** button again.

Inserting and Deleting Frames

The Picture Library allows you to insert and delete frames in a previously created sequence of frames. You can insert frames as well as columns and lines.

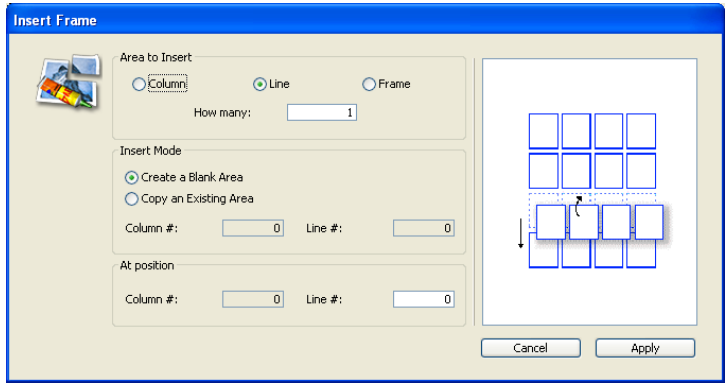
IMPORTANT: *Inserting or deleting frames modifies the original picture. As in the case of manual modification (see the "Modifying Picture Contents" paragraph on page 896), if the original picture includes specific characteristics (vector graphic, alpha channel, and so on), they will be lost when the final sequence of frames is saved.*

► To insert frames:

1 **Select the frame sequence to modify.**

2 **Click on the insertion button in the toolbar of the library.**

The following dialog box appears:



3 **Indicate whether you want to insert a column, line, or frame, and indicate how many in the upper portion of the dialog box.**

The preview area on the right side of the dialog box shows you an example of the selected operation. Note that inserting an element moves the others (no element is replaced).

4 **In the "Insert Mode" area, indicate whether you want the inserted element to be blank or whether it should contain the contents of an existing element.**

In the latter case, you must designate the element to recopy.

Note The number of the first column and first line is 0.

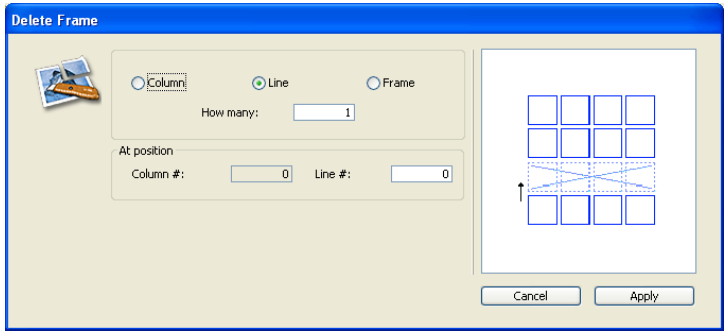
5 **Indicate the location in which you want the new element to be inserted and click Apply.**

The element is then inserted in the picture.

▶ To delete frames:

1 **Click on the deletion button in the toolbar of the library.**

The following dialog box appears:



2 **Define the element (column, line, or frame) that you want to delete and indicate how many.**

The preview area on the right of the dialog box shows you an example of the selected operation.

3 **Indicate the position where you want the element to be deleted and validate the dialog box.**

Note The number of the first column and first line is 0.

Shortcuts for Inserting Picture Buttons and Pop-up Menus

When a picture is defined as a sequence of frames, you can use shortcuts to insert it as a picture button or a picture pop-up menu in your forms (for more information, refer to "Picture Buttons" on page 566 and "Picture Pop-up Menus" on page 579).

- To create a picture button, drag the picture from the library and drop it in the form.
- To create a picture pop-up menu, drag the picture from the library and drop it in the form while holding down the **Shift** key.
- To create a standard picture, drag the picture from the library while holding down the **Alt** (Windows) or **Option** (Mac OS) key.

  Dragging and dropping a picture that is not defined as a sequence of frames will cause it to be inserted as a standard picture.

Note In the Form editor, the Property List allows you to distinguish between two types of pictures: **Picture Library** (dynamically updated when the source picture in the library is modified), and **Static Picture** (not associated with the library's source picture). Refer to the "Dissociating a Picture from its Library Source" paragraph on page 455.

15 Lists

This chapter tells you how to create and use lists. A list is a set of possible values. You can use a list to do the following:

- Provide the user choices from which to select an entry for a field or enterable object,
- Restrict the valid entries to those in the list,
- Exclude the entries in the list from being entered.

4D lets you associate a small icon with each item in a list or hierarchical list. Where appropriate, the small icon is displayed to the left of the item. For example, you can display the small icons in hierarchical lists.

When a list is used as a choice list for a field or enterable object, the user can simply select from the list instead of typing the entry. For example, you may want to create a choice list for entering job titles in a personnel database.

You can also use lists to provide restrictions on data entry. One list may provide the required values for a field, excluding all others. Another list may provide the excluded values for a field, preventing any value in the list from being entered.

You can also create hierarchical lists. A hierarchical list associates a sub-list with each item of the list.

Your lists can offer up to 8,000 choices in a single database and the maximum length of each item is 2 billion characters.

For information about adding a choice list to a field as a field attribute, see the "Allow Choice List" paragraph on page 284. For information about using lists with data entry controls, see the section "Using Choice Lists" on page 492.

Lists are often used in methods. For example, a list is a convenient place to store the elements of an array. An array stores a list of values in memory. You can use lists to store the elements of pop-up menus, hierarchical lists, combo boxes, tab controls, and other multi-valued interface objects. You transfer the contents of the list to the interface object using a method or by assigning the list to the object in the Property List window.

You create 4D lists with the Lists editor found in the tool box. You use the Lists editor to do the following:

- Create lists and hierarchical lists,
- Add items to lists,
- Associate small icons with list items,
- Delete lists,
- Delete items from a list,
- Sort items in a list or sublist,
- Make a choice list user-modifiable,
- Make hierarchical list items editable or a tab control active,
- Specify the spacing between hierarchical list items,
- Insert hierarchical lists or hierarchical pop-up menus in forms.

4D Server Object locking occurs when two or more users attempt to modify the same list at the same time. If a user is modifying a list in the Design environment, the list is locked. Other users cannot modify the list, the list name, or any of the items in the list, until the first user frees the list by closing it.

Designing Lists for Data Entry

One use of lists is to provide the user with a list of values from which to choose during data entry. The following are some considerations about lists that stem from this purpose:

- You can make a list available for every form or for selected forms.
- You can restrict the possible entries to those in the list or you can allow the user to type additional entries.

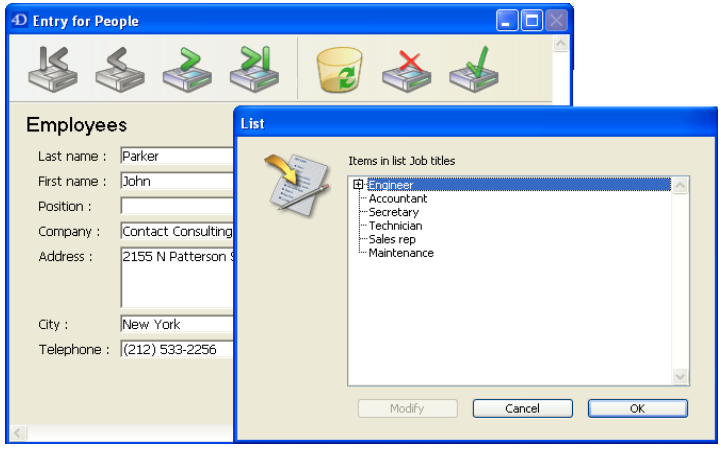
- You can allow the user to modify the list or you can prevent the user from modifying it.

  You can attach a list to a field as one of the field properties. Attaching a list to a field causes the list to appear whenever that field is selected during data entry, whether directly in the list or on the detail page. The user can select an entry from the list. If the list is sorted, the list automatically scrolls as the user types characters at the keyboard. For instance, if the user types “N,” the list scrolls to the first entry starting with “N.” The user can stop typing when the desired choice appears and select it from the list.

  If you attach a list to a field using its Field Properties in the Structure editor, the list will also appear when the field is selected in the Query editor. For more information, see the “Allow Choice List” paragraph on page 284.

  You can also attach the list to the field as a data entry control in a form. The list will appear only when the field is selected in this input form, not in all forms or the Query editor.

  The figure below shows a choice list being displayed.



  If the number of items that the list contains is limited, you may not need to use lists. For instance, in the case of a list that has only two values, such as Male or Female, you may consider using a Boolean field. This would allow you to use interface elements such as radio buttons or picture radio buttons. Even for cases that include four choices, you can use check boxes.

Hierarchical Lists

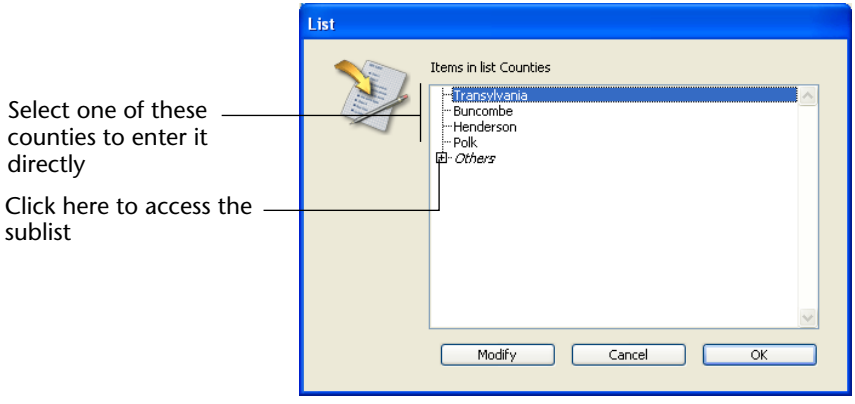
4D allows you to create hierarchical lists. Selecting an item from the parent list displays a sublist.

There is no standard way to use hierarchical lists; it depends on what you use them for. For example, a list of counties for a given state would take many entries, thus slowing down the selection process. There are several approaches when you want to use hierarchical lists for this type of problem.

Besides their use for entry purposes, you can use hierarchical lists to populate the following objects:

- Hierarchical lists
- Tab controls
- Hierarchical pop-up menus
- Scrollable areas
- List boxes,
- Combo boxes.

You can often divide a list of values in two value categories. In the example mentioned above, you could determine which counties are selected the most often. If 80% of the values selected refer to a handful of counties, you can place those counties in a list and place the remainder in a sublist, as shown below.



This allows the user to be able to select an entry directly in 80% of the cases, thus making entry faster.

A different approach consists in organizing counties into different regions. The first list that is displayed allows the user to select the region and the sublists allow him to select the county. In this case, each selection will require the user to select a region, followed by a county. This is still faster than selecting a county from a long list.

Required and Excluded Values

Some data entry tasks are not crucial. If you enter a value that does not appear in a list, it may be perfectly acceptable. However, you may have an application in which an entry must be one of the values in a list. Any different value would cause a serious consequence such as delaying bill payment.

4D allows you to make a list required as part of the data entry controls on a form. This type of data entry control prevents a user from entering any value other than the ones in the list. For example, your company may have a specific group of job titles that are allowed in a personnel database.

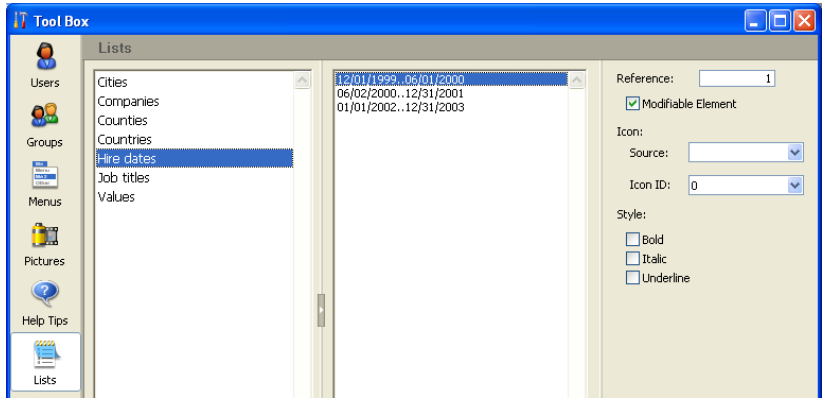
Another data entry control makes it possible to exclude the values in a list. The user then cannot type in a value that should not appear in the field. For example, your company may be prevented from doing business in certain countries. Placing them in an excluded list prevents them from being entered.

Non-Sequential Ranges of Values

One of the most useful data entry controls is the Maximum and Minimum setting for a number, date, or time field. Setting a maximum and minimum value prevents a user from entering a value outside this range.

Suppose you have three acceptable ranges for the field. You can use a list to create such non-sequential ranges. If you then make this list required for a field, values outside the three ranges are not accepted.

The figure below shows a list of ranges:



On the other hand, you could create a list that specifies the ranges that are not valid. If you then make this list an Excluded list for a field, any entry within these ranges is not accepted.

When setting ranges, you should enter two dots ".." between the minimum and maximum values; otherwise 4D will not consider them as ranges but as a string value.

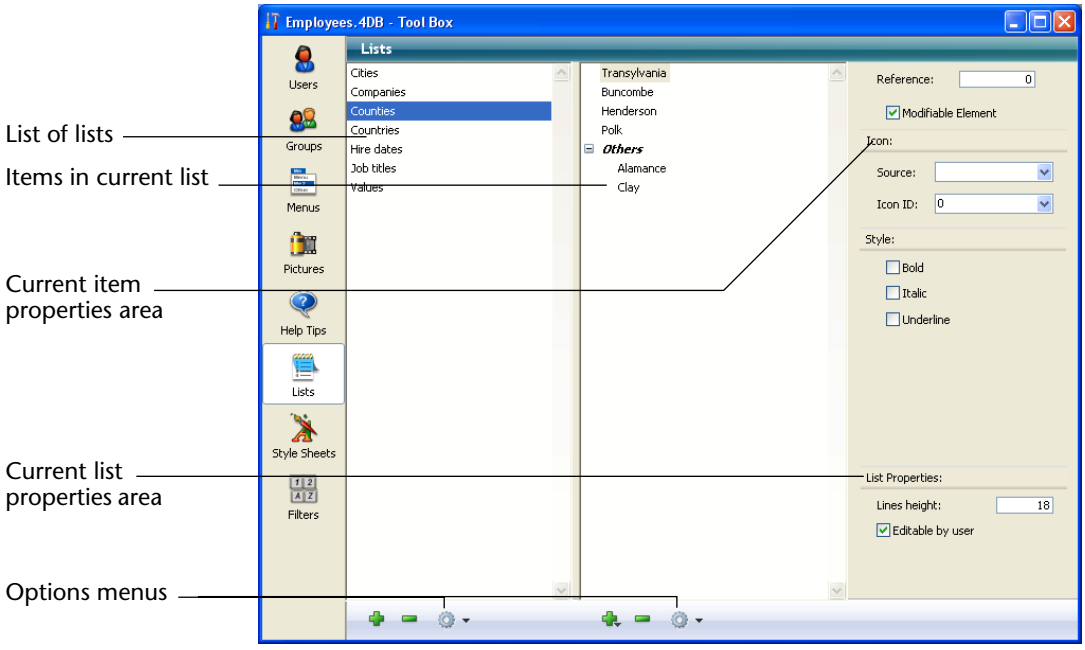
Creating Lists

You create lists with the Lists editor of the tool box. You can modify any list at any time by returning to the Lists editor and making changes.

► To create a list:

1 Choose Tool Box > Lists from the Design menu.

The following window appears (empty by default).



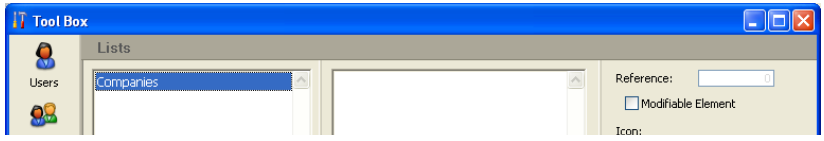
The Lists editor displays the names of existing lists on the left. The central part of the window displays the items of the current list and the right part displays the properties of the current item and the list.

2 Click on the add button located below the list of lists.
OR
Right-click in the list of lists and select the Add command from the contextual menu.

4D creates a new list named *New ListX* by default.

Note If there is already at least one existing list, the **Duplicate** command is available. It can be used to quickly create a new list having characteristics in common with an existing list.

3 Change the name of the list and press Tab to validate your entry.



You can rename a list at any time by selecting the **Rename** command from the contextual menu or from the options menu of the editor, or by using the **Alt+click** (Windows) or **Option+click** (Mac OS) shortcut, or simply by clicking twice on the list you want to rename.

You have created a new empty list. Now, you will create the items that will appear in the list.

Adding Items to Lists

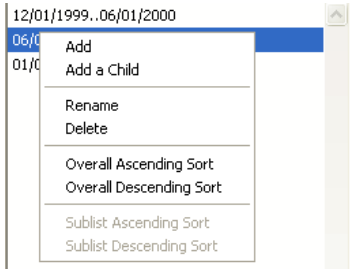
When you are adding items to a list, you can append new items to the end of the list or insert them anywhere in the existing list.

► To append items to a list:

1 **Select the name of the list to which you want to add items.**

If the list already contains items, they appear as a list on the right-hand side of the editor. If you want to insert a value into an existing list, select the value located above the value you want to insert. The new value will be created after that value.

2 **Click on the add button found beneath the list of items.**
OR
Choose Add from the options menu or from the contextual menu of the list of items.

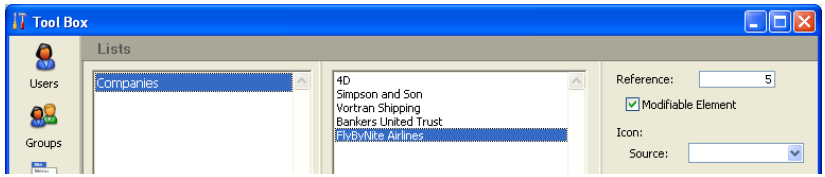


4D creates a new item in the list named *New Item # X*.

3 **Enter the item value and press Tab to validate your entry.**

You can rename a list item at any time by selecting the **Edit** command from the contextual menu or from the options menu, or by clicking twice on the item.

4 **To add other items to the list, repeat steps 2 and 3 as many times as necessary.**



Once the values entered, you can move the list items by drag and drop. You can also sort them by alphabetical order (see the "Sorting a List" paragraph on page 916).

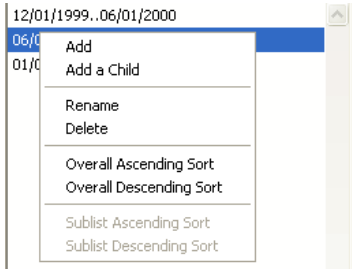
Creating a Hierarchical List

You can add a sublist to any list item. The number of levels of hierarchy is not limited.

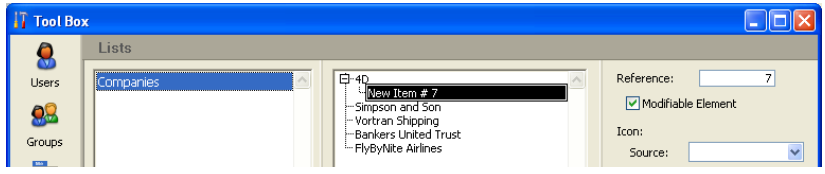
► To create a hierarchical list:

1 **Select the list item to which the sublist will be attached.**

2 **Choose Add a Child from the options menu or from the contextual menu of the list of items.**



4D expands the selected list item and adds a new item (named *New Item # X* by default) for the first item on the sublist.

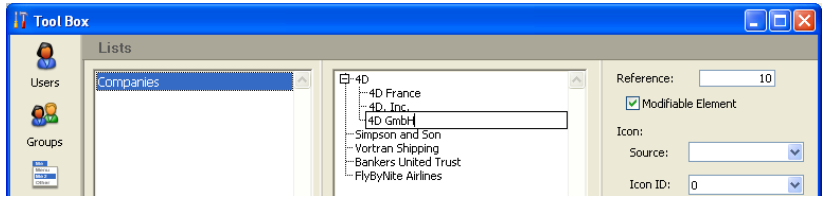


3 **Enter the value of the item.**

You can rename the item of a sublist at any time by selecting the **Edit** command from the contextual menu or from the options menu, or by clicking twice on the item.

4 **To add other values to the sublist, keep the item selected and click on the add button located beneath the list of items.**
OR
Select the parent item and use the Add a Child command again.
If you choose the **Add a Child** command when the item of a sublist is selected, you will create an additional sublevel in the hierarchy (see below).

5 **Repeat steps 2 to 4 as many times as necessary.**



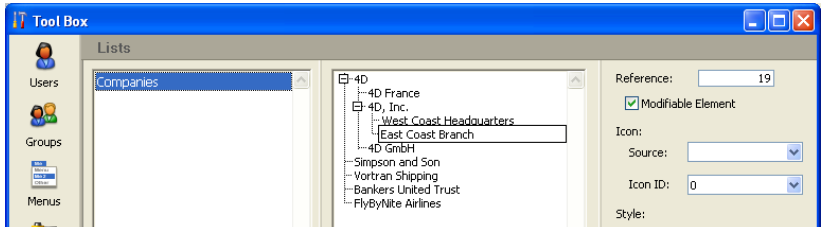
If desired, you can attach sublists to sublist items to continue the hierarchy.

► To attach a sublist to a sublist item:

1 **Select the sublist item.**

2 **Select Add a Child from the options menu or the contextual menu of the list of items.**

3 **Enter the item normally and repeat the process of entering items or attaching sublists to items, as desired.**
The following illustration shows a three-level hierarchy.



Deleting Items and Lists

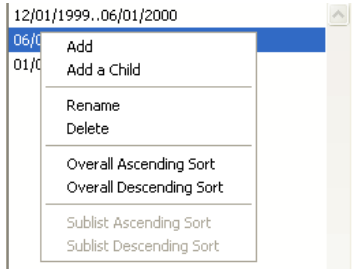
You can delete items at any level of the hierarchy. Keep in mind that it is not possible to cancel the deletion of an item or list.

► To delete an item:

1 **Select the list that contains the item you want to delete.**

2 **Select the item you want to delete from the current list area.**
If necessary, expand the list.

3 **Click on the deletion button located below the list of items for the current list.**
OR
Select the Delete command from the contextual menu of the list of items.

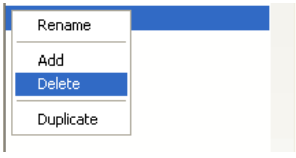


4D deletes the item from the list.

▶ To delete a list:

1 **Select the list you want to remove.**

2 **Click on the deletion button located below the list of lists.**
OR
Select the Delete command from the contextual menu of the list of lists.



4D displays a dialog box that lets you confirm or cancel the operation. If you validate this dialog box, the list will be deleted.

Adding a Small Icon to an Item

You can associate a small icon with an item in a list. When the list is displayed in a scrolling area or on a tab, this icon appears to the left of the value.

The small icons that you use are stored either in the picture library, or in 'cicn' or 'pict' type resources.

Note The "resources" can be data of any type (text, pictures, etc.) that are structured in a format which is defined and used by the program. This concept originates from the Mac OS environment but it can also be used by 4D under Windows. For more information, refer to the "Resources" section of the *4D Language Reference* manual.

▶ To associate a small icon with a list value:

1 **Select the list then the value with which you want to associate the icon.**

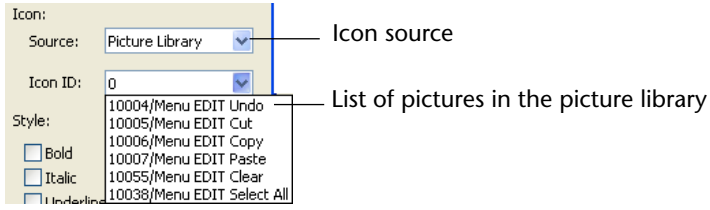
2 **Choose an option from the Source menu of the Icon area.**

This menu is used to indicate the source of the icon. You can choose from the three following options:

- **'cicn' Resource**: cicn type resource
- **'pict' Resource**: pict type resource
- **Picture Library**: Picture stored in the picture library.

When you choose an option, the contents of the **Icon ID** menu are updated in order to list all the pictures available in the database that correspond to the type set in the Source menu.

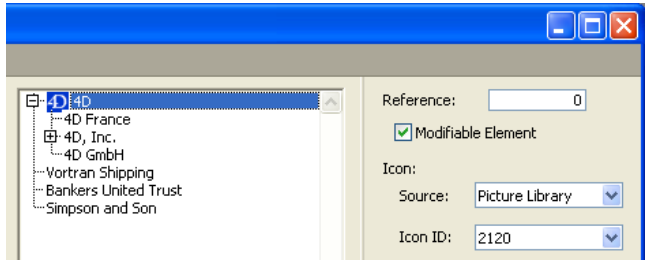
3 **Choose a value from the Icon ID combo-box.**



The icons are listed in number/name form. If you have chosen a resource in the Source menu, the Icon ID menu contains all the corresponding resources that are available to the application.
If you have chosen the "Picture Library" option, the Icon ID menu contains all the pictures stored in the library.

Note For more information about the picture library, refer to the chapter "Picture Library" on page 891.

The icon is added to the left of the label in the list of items:



Note Depending on the size of the icons you use, you may need to modify the list's height. For more information, refer to "Setting the Minimum Height of a List" on page 919.

To delete an icon reference, select the value in the "Icon ID" combo-box and hit **Backspace** or enter 0.

Adding a Reference to an Item

The current item properties area contains an entry area for the item's Reference. The Reference is designed as a unique ID for the item. It is of use only when you manage lists using methods.

When you need to use the language to determine which item in a list a user selects (e.g., which item in a hierarchical menu is selected), you can identify the user's choice using the Reference of the item. For more information, see the section "Hierarchical Lists" in the *4D Language Reference* manual.

Specifying Ranges in a List

4D allows you to enter ranges of numbers, dates, and times in a list. You can use these ranges as data entry validation ranges by making the list required or excluded in a form.

► To create ranges in a list:

1 **Create the list you want to use for ranges.**

2 **For each item, enter the minimum value of the range, two periods (..), and the maximum value.**

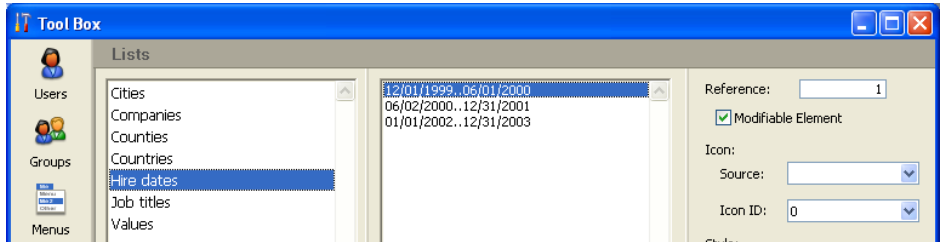
For example:

100..150

sets the range between 100 and 150, inclusive.

3 **Continue specifying ranges as separate items until you have set as many as you need.**

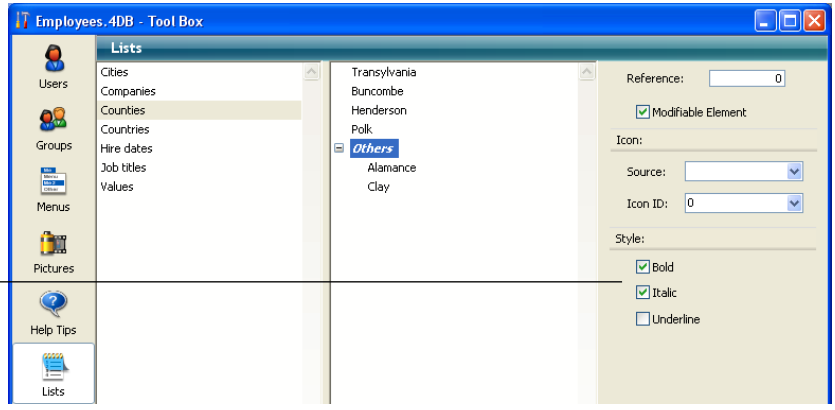
Here is an example of a list using ranges:



Specifying Font Attributes

When a list is used as a choice list, you can display list items in bold, italic, and/or underline.

To apply styles, select the value in the list and check the **Bold**, **Italic** and/or **Underline** options as desired. You can choose several options if you want to combine the styles. The following illustration shows the bold and italic attributes applied to a list item.



When the list is used as a choice list, the selected style attributes will be used, as shown in the following illustration.

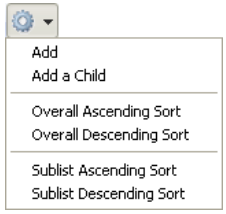
Sorting a List

4D keeps the list of items in the order in which you enter them. You can sort the list or sublist alphabetically so that entries are more easily accessible to database users. Since a sorted list automatically scrolls to match characters typed at the keyboard, sorting usually makes data entry easier.

► To sort a list or sublist:

1 **Select the list that contains the choices you want to sort.**

2 **Choose one of the sort options available from the contextual menu or from the options menu located below the list of items:**



Sort Options

The following sort options are available:

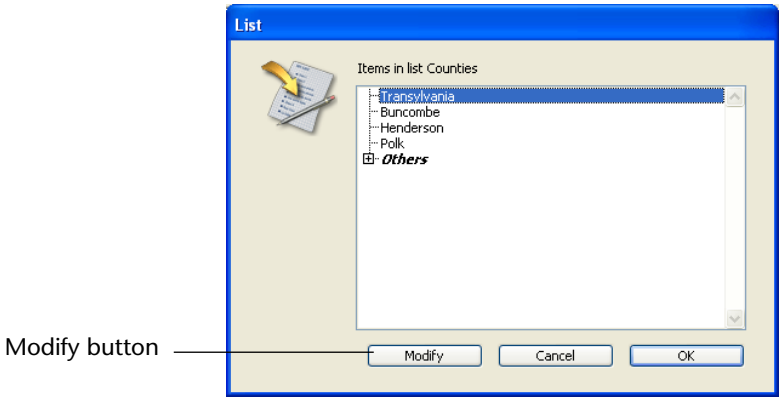
- **Overall Ascending Sort**: 4D sorts the list and any sublists by ascending order (from A to Z).
- **Overall Descending Sort**: 4D sorts the list and any sublists by descending order (from Z to A).
- **Sublist Ascending Sort**: 4D sorts only the sublists by ascending order.
- **Sublist Descending Sort**: 4D sorts only the sublists by descending order.

Making a Choice List Modifiable

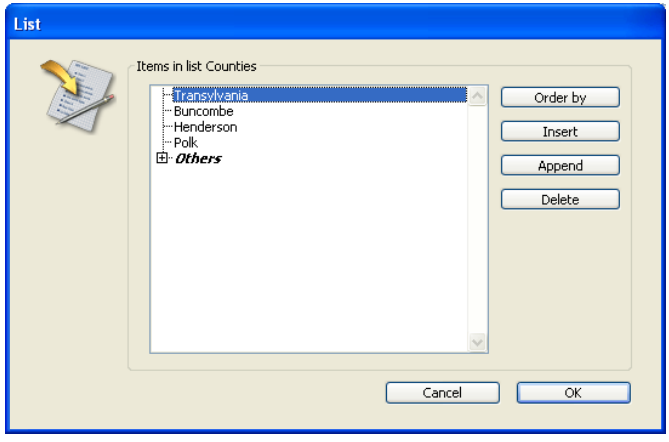
4D allows you to specify whether a list of items can be changed by the user when it is displayed as a choice list. By default, a list is not modifiable.

If you allow a list to be user modifiable, the user has access to a special List editor when using the database. The special List editor is for the assigned list only. The user cannot add lists, delete lists, or change any other list. If a list is modifiable, the user can make any necessary changes to that list's items.

If a list is modifiable, the **Modify** button is enabled in the List dialog box when using the database.



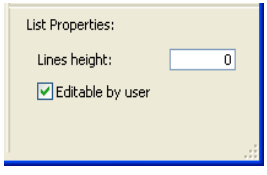
When the user clicks this button, the "user" List editor is displayed. The following shows this List editor:



► To make the list user-modifiable:

1 **Select the list that you want to make modifiable.**

2 **Check the Editable by user option in the "List Properties" area:**



The list can then be modified by the users of the database.

To prevent the user from modifying a list, select the list and deselect the **Editable by user** option.

Setting the Minimum Height of a List

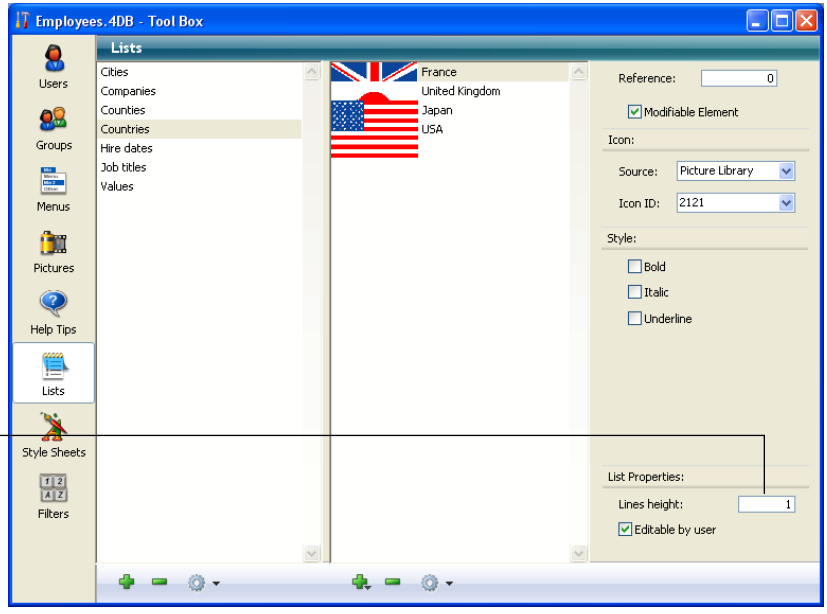
When 4D displays a list as a choice list, it uses the font size of the hierarchical list object to determine the vertical spacing between adjacent list items. If you use a list to specify the values of a hierarchical list, you can specify a larger vertical spacing. The main reason you would want to do this is to provide additional space for icons that are attached to list items. Or, you can use this feature simply to spread out the list items.

► To specify a minimum height:

■ **Enter a value in points in the "Lines height" entry area.**

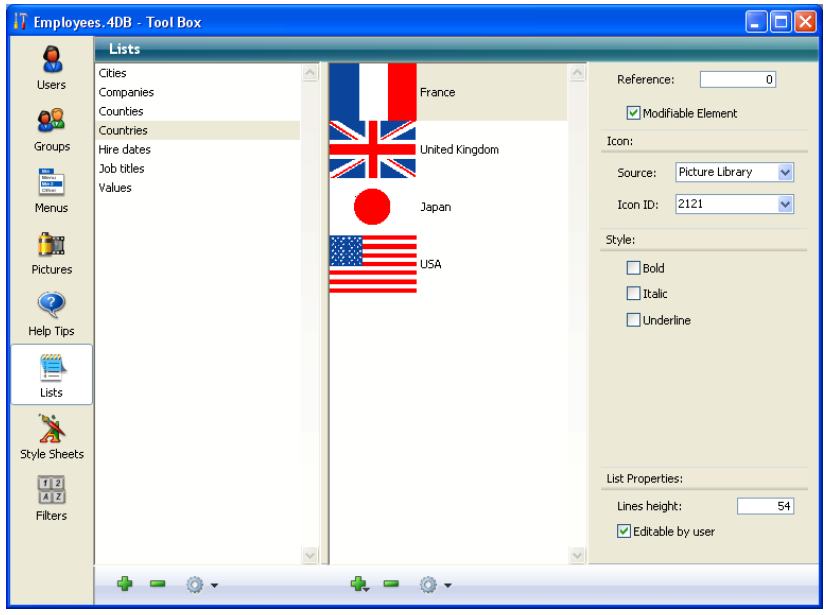
The effects of this value are displayed immediately in the list of items area. The following illustrations show the effect of increasing the minimum height.

Minimum Height: 1 (default value)



Minimum height entry area

Minimum Height: 54

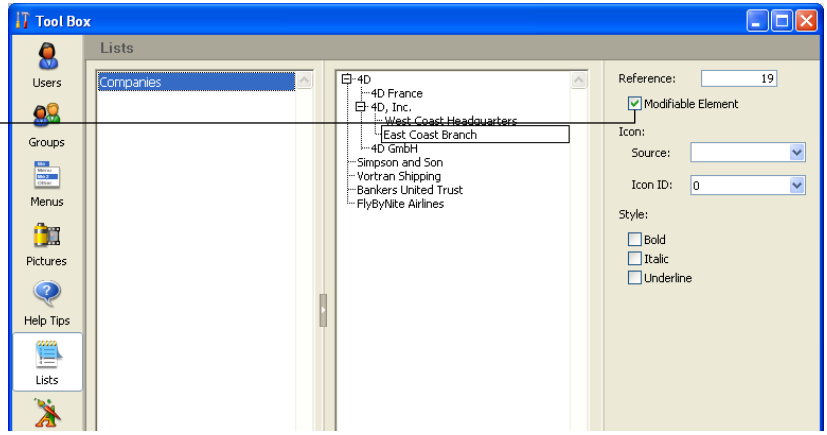


Modifiable Element Option

The Lists editor offers the **Modifiable Element** option for each element of a list. It is checked by default.

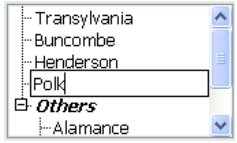
This option is only used with lists displayed in the form of **hierarchical lists** or **tab controls**. Its action is different in both cases.

Modifiable Element option



Hierarchical Lists

A list can also be used to specify the items in a “hierarchical list” object. When the list is used in this manner, you can control whether each item in the list can be edited by the user. If a list item is modifiable, the user can hold down the **Alt** key under Windows or **Option** key under Mac OS and click on the item (or simply click twice) to get an insertion point. An editable item in a hierarchical list is shown below.



In this case, the **Modifiable Element** option lets you allow the user to modify the element of the hierarchical list.

Tab Controls

When a list is associated with tab controls, you can enable or disable each tab control that corresponds to an item of the list. A disabled tab control will be displayed in gray in the form. In the following example, the tab control “Henderson” is disabled.

In this case, the **Modifiable Element** option lets you enable the tab control corresponding to the element.

Dragging a List into a Form

4D allows you to use shortcuts to create hierarchical lists or drop-down hierarchical menus that are associated with lists. These operations can be performed by dragging lists from the Lists editor.

- To insert a hierarchical list in a form, drag the list from the Lists editor to the form.
- To insert a hierarchical menu in a form, drag the list from the Lists editor to the form while pressing the **Shift** key.

16 Style Sheets

This chapter describes the creation and use of style sheets. A style sheet groups together a combination of font attributes — the font type, its size and its style. The style sheets that you set up can be used to set the font attributes for objects in the Form editor.

Each style sheet saves separate sets of font attributes for each of the platform interfaces supported by 4D: Windows XP, Windows Vista, Mac OS X and Windows 2000. For example, for the "Buttons" style sheet, the Mac OS platform interface could use Lucida Grande as the font, while the Windows platform interfaces could use Tahoma. Similarly, the font sizes can be specified separately for each platform interface.

In addition to harmonizing the interface of your applications, the use of style sheets brings three major advantages:

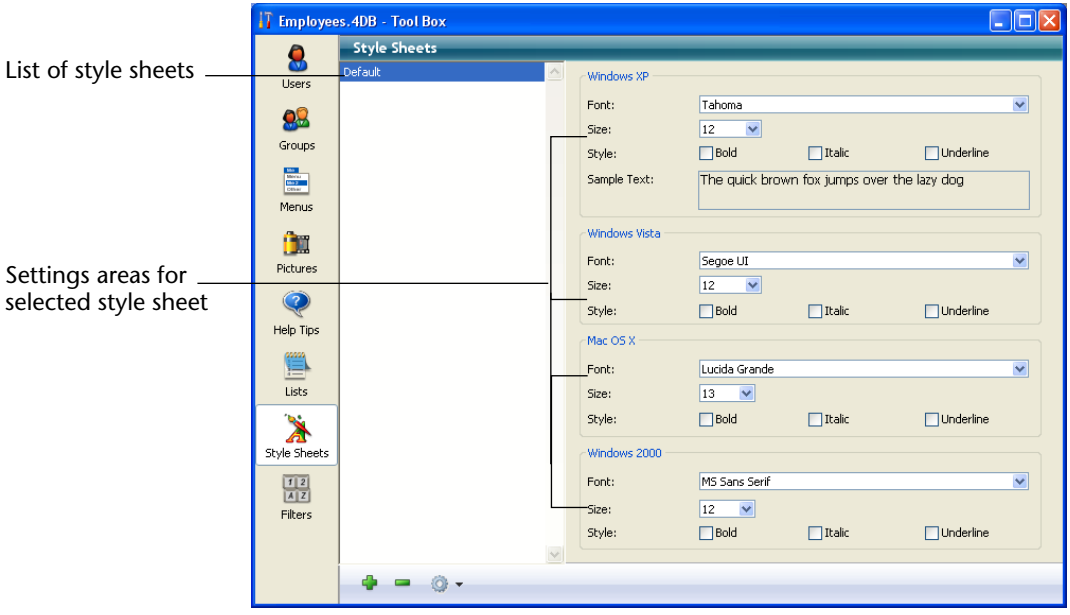
- Saves time during development: For each object, you specify a group of settings in a single operation.
- Facilitates maintenance: Style sheets modify the appearance of all the objects that use them. Changing, for example, the font size in a style sheet will change the font size for all the objects that use this same style sheet.
- Controls multi-platform development: When a style sheet is applied, 4D automatically uses the parameters set for the platform on which the form is displayed (if the object has the System appearance property).

Creating a Style Sheet

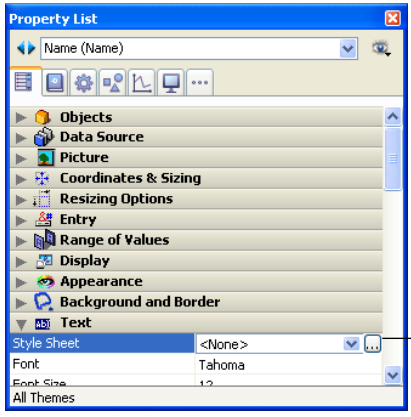
You can create style sheets using the Style Sheet editor found in the 4D Tool Box.

► To create a style sheet:

1 **Choose the Tool Box > Style Sheet command in the Design menu.**
OR
Display the Tool Box and click on the Styles button.
The style sheets page of the tool box appears:



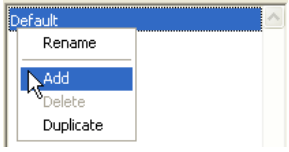
You can also open this window from the Form editor by clicking on the [...] button located next to the style sheet drop-down list of the Property List.



The order in which the areas appear depends on the current operating platform of 4D: for example, the **Mac OS X** area (as well as the associated sample text) is displayed at the top of the page when 4D is running under Mac OS X.

By default, only the "Default" style sheet is available. You can modify this style sheet. However, keep in mind that it is used by default when creating objects in the Form editor.

2 **Click on the add button**  **of the editor.**
OR
Right click in the Style Sheets list area and choose the Add command in the contextual menu:



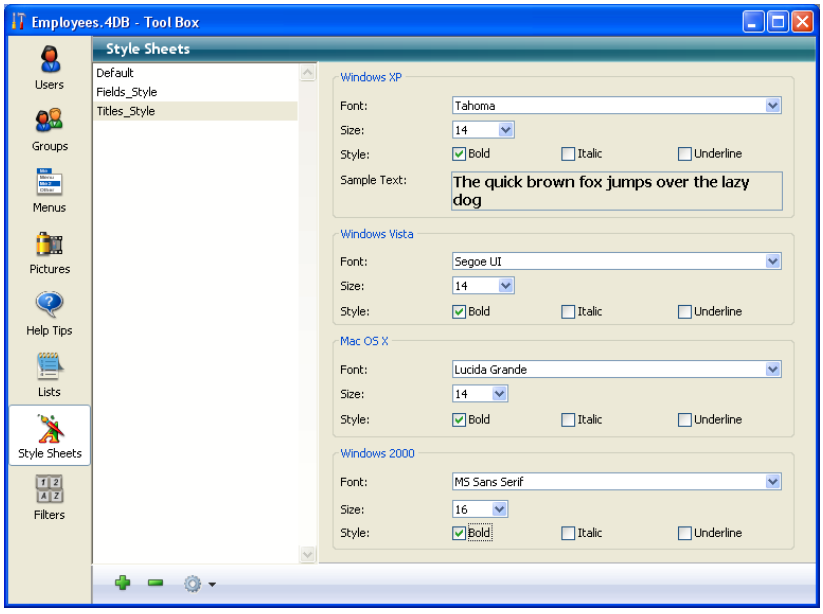
A new style sheet is created in the list. Its name by default is "Style sheetX."

3 **Click on the style sheet so that it switches to edit mode and give it a new name**

Once the name has been validated, the list of style sheets is automatically re-sorted by alphabetical order.

4 **In the style sheet settings area, choose the desired font as well as its size and style options for each operating platform of the database.**

The modifications are saved automatically. The “Sample Text” area reflects your changes for the current platform.



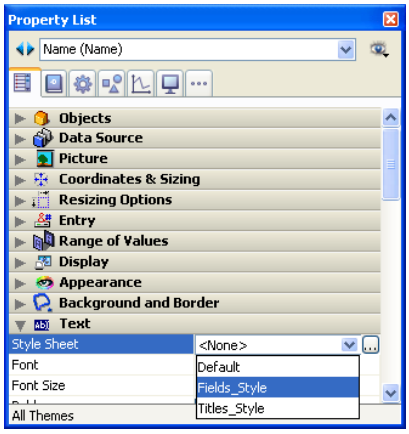
If you want, you can create a new style sheet by duplicating an existing one. This way you avoid having to reset the points in common between the new style sheet and the one duplicated.
To duplicate a style sheet, click with the **right button** of the mouse on the name of an existing style sheet and choose the **Duplicate** command in the contextual menu. You can also select an existing style sheet and choose **Duplicate** from the options menu of the page.

Applying a Style Sheet

Applying style sheets that you have specified lets you avoid having to configure each attribute separately.

The style sheets that have been set up can be used in the Property List of the Form editor. Style sheets can be applied to any object, whether static or dynamic, that includes text: buttons, variables, fields, tabs, hierarchical lists, list boxes, etc.

To apply a style sheet to an object, select the object then choose the style sheet from the Style Sheet pull-down list in the “Text” theme of the Property List:



Your selection sets the font, font size, and font style attributes for the currently selected label or object.

17 Processes

You can increase the functionality of a database by taking advantage of 4D's multi-tasking capabilities. In a multi-tasking system, database operations can run in separate *processes* — separate 4D tasks — that operate independently and concurrently.

Multiple processes are executed at the same time, allowing 4D to carry out several operations simultaneously. For example, one process might print a selection of records while another process allows a user to enter new records. In custom applications, multiple processes are used to manage a multi-window interface. Each window is managed by a different process. The process has its own current selection of records — even if two processes access data in the same table. The number of processes that can be executed at the same time is limited only by available memory.

This chapter explains how to do the following:

- Start a new process,
- View process information including the name, number, status, and total amount of time used by each process,
- Control process execution,
- Control whether a process is visible to users,
- Specify the frontmost process,
- Debug a process.

Processes

A process can be thought of as a 4D task that is carrying out some action — searching for certain records, printing a selection, allowing a user to enter data, and so on. The exact operation that the process performs depends on the method with which it is connected.

As a separate 4D task, each process has the following elements for data management:

- A current selection for each table,
- A current record for each table,
- Process variables,
- Locked records.

In addition, each process can have the following elements:

- Current input and output forms for each table,
- A menu bar,
- One or more windows,
- One active window (the frontmost window).

For a complete list of process elements and information about creating windows and menu bars, refer to the *4D Language Reference* manual.

You may notice that all of these elements are found in the 4D Application environment. All processes have the same basic properties as the Application environment and allow you to perform the same operations that you can in the Application environment.

However, instead of having to perform the operations directly in the Application environment, processes allow you to use methods to specify the actions you want 4D to take.

Anything that can be done with the 4D language — any user interface that can be created or operation that can be performed — can be done from a process.

Having multiple processes open at once gives you the ability to perform different actions or work with different aspects of a database simultaneously.

Opening multiple processes allows the user to do the following:

- **Work with more than one active window**: You can have several active windows open at the same time. For example, you can enter data in one window and receive messages from colleagues in another window.
- **Work with more than one current selection at a time**: Because each process has its own current selection, each process can have a different current selection from the same table. For example, an Employees database might contain a list of employees and their occupations. In a single-process database, you could display all engineers at once, or all accountants at once, but you could not make both selections the current selection for the same table. In a multi-process database, you can display the records of all engineers in one process and the records of all accountants in another process.
- **Work with more than one current record at a time**: Each process can have a different current record. For example, you might want to compare one employee to another using an input form. In a single-process database, you can display only one employee's record. In a multi-process database, you can display each employee's record in a different process.
- **Start a lengthy operation in a separate process**: You can perform a time-consuming operation such as printing a large selection of records in a separate process while you continue to work on your database.
- **Work with more than one input or output form at a time**: You can view data in several different forms at once. For example, you could display a selection of records in a standard output form in one process and in a special report form in another process.

Processes Created and Managed by 4D

4D automatically creates and manages the following processes which control the operation of 4D:

- **Application**: This process controls the Application environment.
- **Design**: This process controls the Design environment. This process does not exist in compiled mode or in a database opened with 4D Desktop.
- **Internal Timer Process** and **Internal Bridge Process:** This process manages internal controls needed for program functioning.
- **4D Compiler**: This process manages the compiler integrated into 4D. For more information about this point, refer to the chapter "Compilation" on page 1229.

- **$Runtime Explorer**: This process manages the Runtime Explorer window when it is displayed as a floating palette. It is created as soon as the Runtime Explorer is opened.

In addition, 4D creates and manages the following processes:

- Indexing process,
- On Event Manager process,
- Web Server process.

Unlike user-created processes, the processes created by 4D are always running and cannot be frozen or aborted. For more information about viewing processes in the Runtime Explorer, see the section "Using the Process List" on page 937.

For more information about the processes created by 4D, refer to the *4D Language Reference* manual.

Time-Sliced Execution

Since in reality more than one process cannot execute at the same time, when you open multiple processes, 4D slices the total processing time so that execution is divided between all open processes. Execution alternates between processes so rapidly that the processes appear to be executing simultaneously. For instance, processing time is split between the Design process, the Application process, and the Cache Manager so that some milliseconds might be devoted to the Application process, the next to the Design process, the next back to the Application process, and so on.

Starting a New Process

4D allows you to start your own processes from the Design and Application environments.

Each process that you start can perform a different task or present a different aspect of the data contained in your database.

The functionality of the process can be enhanced by a user interface created using any of the editors in the Design environment or using the 4D language. For instance, you can display an input form in a process to allow a user to enter records.

► To start a new process:

1 Create a method.

The specific operation that each process performs depends on the commands and functions in the method. For more information about the 4D commands, refer to the *4D Language Reference* manual.

2 Specify that 4D should start a new process when the method executes.

You can tell 4D to start a new process in the following three ways:

- Using the New process command in another method,
- Using the Menu Bar editor,
- Using the Method editor or the Execute Method dialog box.

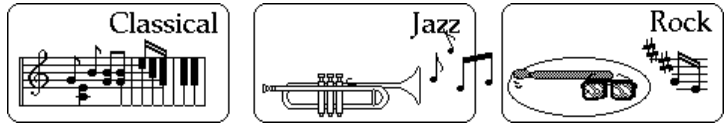
The alternative you select depends on what you are trying to do. Each alternative is described in detail in the following sections.

Starting a New Process Using the New Process Command

There are many circumstances in which you might want to start a new process with the New process command. For instance, you might want to start a new process when a user clicks a button. You can do this by starting the process in the button's object method.

When you start a new process using the New process command, you should place the New process command in the method from which you want to start the new process. When that method executes, 4D starts a new process for the method specified in the parameters of the command and continues executing the original method.

The figure below shows a set of buttons in a compact disc database.



The object method for the **Jazz** button uses the New process command to start a new process for the *JazzSearch* method:

```
myProcess:= New process ("JazzSearch";64000;"JazzRecords")
```

The new process, myProcess, carries out the actions specified in the *JazzSearch* method.

The *JazzSearch* method creates a selection of Jazz compact discs and displays them in a window. It contains the following statements:

```
QUERY ([Compact Discs];[Compact Discs]Category="Jazz")
If (Records in selection ([Compact Discs])>0)
```

```
   RefNo:=Open window (50;50;300;250;8)
   MODIFY SELECTION ([Compact Discs])
   CLOSE WINDOW
End if
```

When a user clicks the **Jazz** button, 4D opens a window and displays all the compact discs that contain Jazz music.

The window is running in the new process, myProcess.

If a different button is clicked, another process is started. If the user clicks the **Jazz** and **Rock** buttons, 4D starts two processes and displays the selection for each in its own process window.

The user can double-click records in either window to modify them in the current input form for that process.

For more information about the New process command, refer to the *4D Language Reference* manual.

Starting a New Process Using the Menu Bar Editor

You will often find that you want a new process to start whenever a user chooses a menu command. For instance, you might have a menu command that prints a group of records. Since printing records can be time-consuming, you specify that 4D should start a new process whenever that menu command is selected.

► To start a new process using the Menu Bar editor:

1 **Choose a menu bar in the Menu Bar editor.**

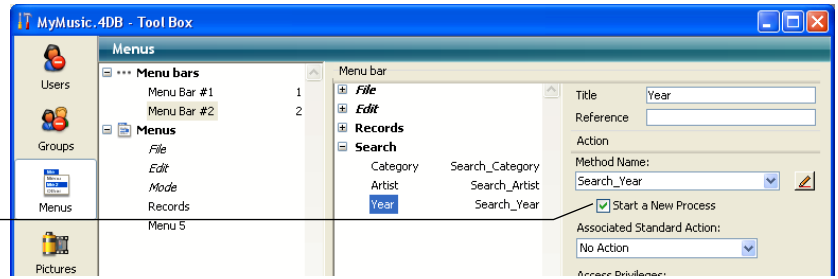
This editor is located in the tool box.

2 **Choose a menu from the list of menu bars then select the menu command for which you want to start a new process.**

3 **Click the Start a New Process check box.**

This specifies that a new process should be started whenever that menu command is selected.

The following illustration shows the **Start a New Process** option selected for the **Year** menu command:

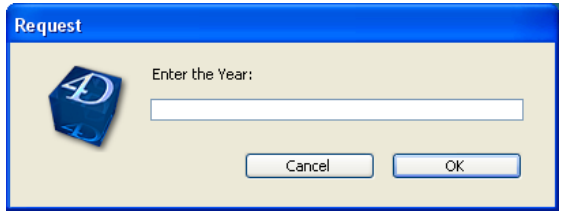


Start a New Process check box

The **Search** menu allows users to perform various searches. In the above example, the method for the **Year** menu command allows the user to enter the year. It then searches for all compact discs produced in that year and displays the selection in a window.

```
vYear:= Request ("Enter the Year:")
If (OK=1)
   QUERY ([Compact Discs];[Compact Discs]Year=vYear)
   If (Records in selection ([Compact Discs])>0)
      RefNo:=Open window (50;50;300;250)
      MODIFY SELECTION ([Compact Discs])
   End if
End if
```

When the user chooses **Year** from the **Search** menu in the Application environment, a dialog box appears requesting that the user enter a year.



When the user clicks the **OK** button, 4D displays the selection of compact discs produced the year the user entered. If the user chooses **Category** from the **Search** menu, the user can perform another search based on the type of music the compact disc contains.

Starting a New Process by Executing a Method

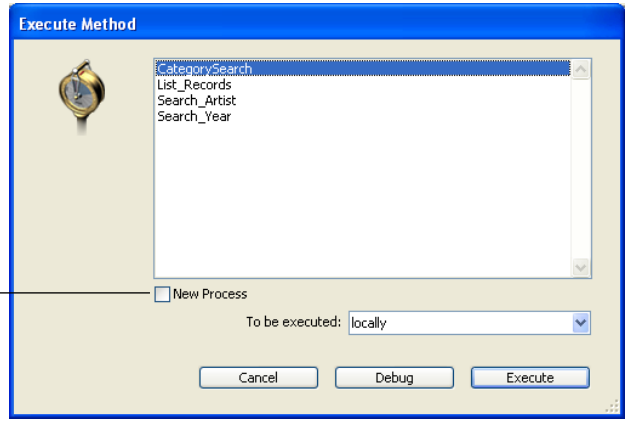
You can choose to start a new process when you execute a method using the Execute Method dialog box or directly from the Method editor. This lets you test the operation of a method in multi-process conditions.

4D Server You can choose to execute a process on 4D Server or on a 4D Client machine. Executing a process on the server lets you centralize the processes that do not require user intervention on the server machine. Executing a process on a specific client machine lets you allocate the processing. For more information, refer to the *4D Server Reference* manual.

► To start a new process in the Execute Method dialog box:

1 **Select Method... from the Run menu of 4D.**
The Execute Method dialog box appears.

2 **Select the method for which you want to start a new process.**
The figure below shows the *CategorySearch* method being selected:



New Process check box

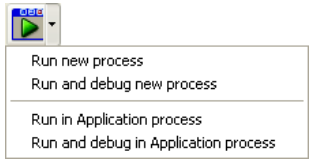
3 **Click the New Process check box.**
Clicking the **New Process** check box tells 4D to start a new process when the method executes.

4 **Click the Execute button to execute the method.**
If you want to monitor or debug the execution of the method, click **Debug**.
4D executes the method within a new process.

To launch a new process from the Method editor:

1 **Open the method to be executed in an editor window.**

2 **In the menu associated with the execution button, choose Run new process or Run and debug new process.**



4D then launches the method in a new process.

Using the Process List

4D lists processes on the Process page of the Runtime Explorer editor window as soon as they are started. Each process is given a process ID, which is the same as the process number (discussed in the next section). You use this process ID to identify a specific process in commands and functions.

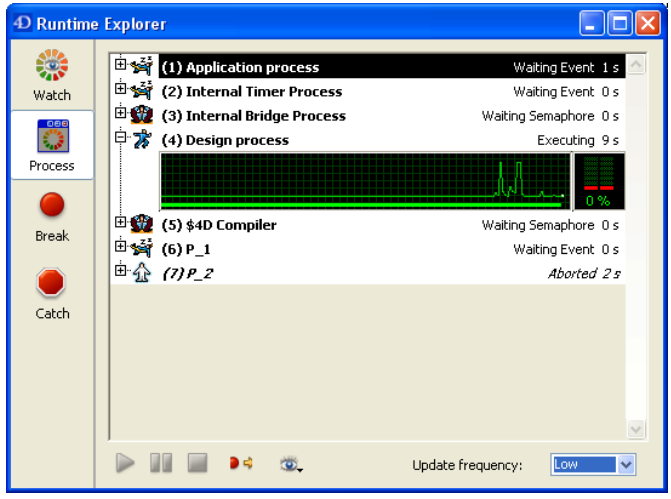
► To view the Process List:

1 **Display the Runtime Explorer Window.**

For more information, refer to "Runtime Explorer" paragraph on page 136.

2 **Click the Process button.**

Several processes are shown by default (see the "Processes Created and Managed by 4D" paragraph on page 931).



For each process, the Runtime Explorer editor window gives the following information:

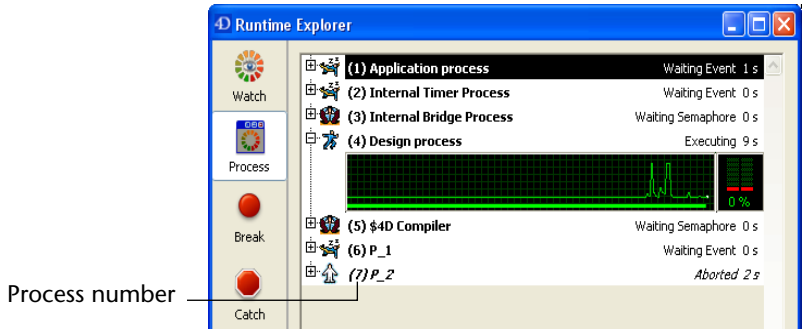
- Process number,
- Process name,
- Current status of the process,
- Total amount of execution time in seconds that the process has taken since it was started,
- The graphical representation of the status and CPU time allocated to a process. For each process you can hide or display its graph by clicking on the icon located to the left of the process name.

The process attributes are explained in detail in the following sections.

4D Server 4D Client's Process page of the Runtime Explorer controls processes for a particular client. 4D Server's Process page of the Runtime Explorer controls processes for all clients connected to the server. For more information, refer to the *4D Server Reference* manual.

Process Number

The default processes (Application process, Internal Timer Process, Internal Bridge Process, Design process and Web Server process) are, respectively, numbers 1, 2, 3, 4 and 5[1].



When you start your own process, it either appears as the next process in sequence or takes the place of a process that has been aborted.

For example, suppose processes 7 and 8 are executing. If process 7 is aborted, the next process to be started becomes process 7.

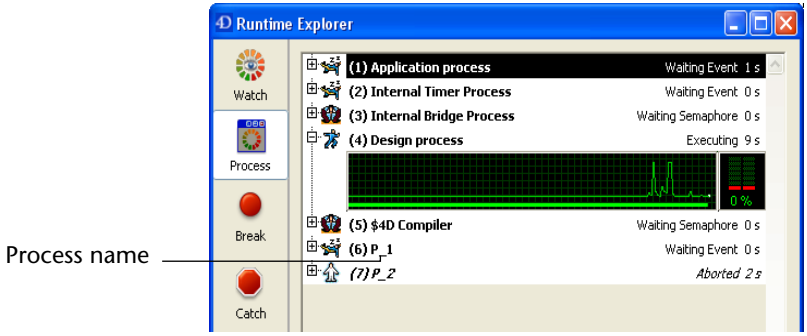
Note Processes are automatically aborted upon completion. You can abort a process before it has completed using the Runtime Explorer. For more information about aborting a process, see the section "Aborting a Process" on page 944.

Process Name

If you start a new process using New process, you can specify its name as a parameter to the New process function. The name specified in the parameter appears as the process name in the Runtime Explorer.

1. If the Web server is launched at startup (see the Preferences), it then becomes number 4 and the Design process is number 5.

For more information about the New process command, refer to the *4D Language Reference* manual.



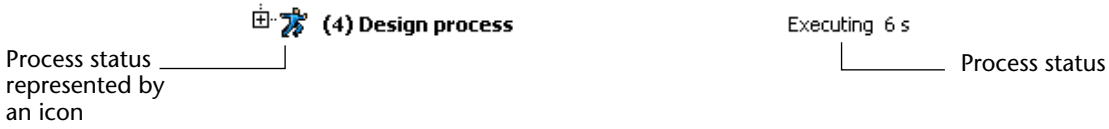
If you do not specify a process name using the New process command, 4D automatically assigns the process a default name. Default names are based on the method used to start the process, as follows:

- **Processes started from a menu command**: If you start a process from a menu command, the process is given the default name "M_*ProcessNumber*." For instance, if process number 7 is started when a menu command is selected, the process is given the name "M_7."
- **Processes started from executing a method**: If you start a process from the Execute Method dialog box or directly from the Method editor, the method is given the default name "P_*ProcessNumber*." For instance, if process number 8 is started programmatically, the process is given the name "P_8."
- **Processes started using New process, but not explicitly named**: If you start a process using a method but do not specify the name as a parameter to the New process command, the process name is left blank.

Note If the name of a process begins with a dollar sign ($), it is a local process that does not have access to tables or 4D Server.

Process Status

The status of a process is the current state of its execution — what the process is actually doing. In the Runtime Explorer, the status of the process is indicated by the icon located to the left of the process name and by the text located next to the process name.



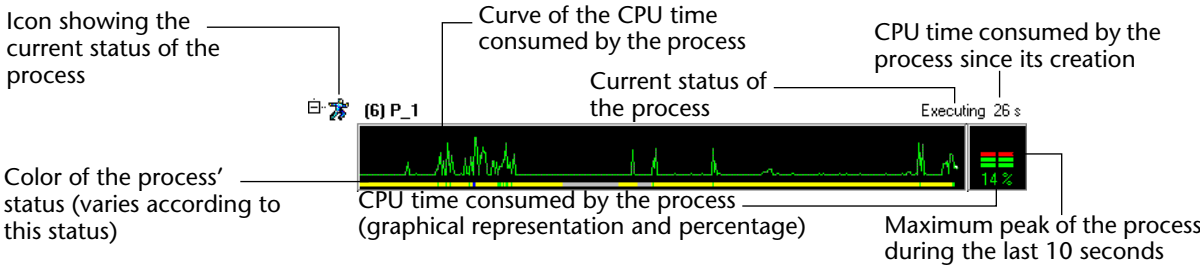
The following is an explanation of each status that can appear in the Runtime Explorer window.

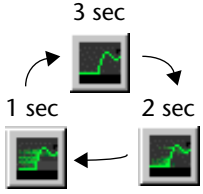
- **Executing**: The process is currently executing.
- **Delayed**: The process is delayed for a specific amount of time. During the period that the process is delayed, it does not take up any processing time. For information about how to delay a process, refer to the *4D Language Reference* manual.
- **Waiting Event**: The process is waiting for an action from the user such as a button being clicked or a menu command chosen.
- **Waiting I/O**: The process is waiting for some input or output to occur. For example, a process might need to wait while a group of records is being updated to disk.
- **Waiting Semaphore**: The process is waiting for the internal processes to finish executing 4D database tasks.
- **Paused**: The process is paused until you tell it to resume execution. During the period that the process is paused, it does not take up any processing time. For more information, see the section "Pausing and Resuming a Process" on page 943.
- **Aborted**: The process has been terminated. When a process is aborted, 4D frees any locked records, cancels any transactions opened by the process but not yet validated or canceled, and frees the current selection and current record. Processes are automatically aborted upon completion. You can also abort a process before it has completed by using the Runtime Explorer. For more information about aborting a process, see the section "Aborting a Process" on page 944.
- **Hidden Modal Dialog**: A process which was displaying a modal dialog box has been hidden so that the user can no longer view the dialog box. The process will remain in this state until the dialog is shown again.

Process Time

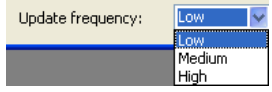
In managing processes, 4D divides processing time among existing processes so that no single process is executing at every moment. Thus, the process time is the total amount of execution time a process has taken (in seconds) since it started executing. Note that the process time does not reflect the total amount of time that has elapsed since the process started executing since, in reality, execution alternates between all open processes.

The Runtime Explorer displays the processing time for each process. If you expand the process graph, it displays the following information:





You can modify the frequency at which data should be updated — which can be every one, two, or three seconds. To do so, choose a value from the "Update frequency" menu found at the bottom of the window.



The greater the update time is, the more CPU time is consumed by the Runtime Explorer process. The number of processes to graphically represent on screen also influences the CPU time consumed by the process.

Note No CPU time is consumed for a process when its graphical representation is closed.

When you click in the graphical area, a vertical line appears where you clicked and a tip indicates the state of the process at that instant. By holding down the mouse button and moving it from side to side, you can view the changes in the status of the process.

The process management commands are accessible by using the toolbar's buttons in the window: **Execute** , **Pause** , **Abort** , **Trace** , **Hide** , **Show** , **Bring to Front** .

Controlling Process Execution

The Runtime Explorer allows you to control the execution of processes by pausing, resuming, or aborting a process. You can also choose to run a process in debug mode. Lastly, you can hide and redisplay its windows at any time.

These operations are covered in detail in the sections below.

Note You can also delay a process for a specific period of time. For more information about delaying a process, refer to the *4D Language Reference* manual.

Pausing and Resuming a Process

You can temporarily suspend the execution of a process by pausing it. You may want to pause a process to give other processes more execution time or to allow an event upon which the process depends to occur.

For instance, suppose you start a process that prints a selection of records. You then realize that you want to modify the data in one of the records so you first pause the process, finish your modifications, and then resume the process to continue printing the records.

► To pause a process:

1 **Select the process on the Process page of the Runtime Explorer.**

2 **Click the Pause button**  **.**

The status of the process in the Runtime Explorer window automatically changes to “paused.” The process remains paused indefinitely until you tell it to resume execution.

► To resume execution of a process:

1 **Select the process on the Process page of the Runtime Explorer.**

2 **Click the Resume button**  **.**

The status of the process returns to the status it had at the time it was paused. For example, if the process was executing before it was paused, the process begins executing again. If the process was waiting for an event before it was paused, it continues waiting for an event.

Aborting a Process

A process is automatically aborted upon completion. However, you may need to abort a process before it completes for debugging purposes. Processes should not be aborted for any other reason. To stop the process from continuing execution, you abort the process in the Runtime Explorer.

When a process is aborted, 4D frees any locked records, cancels any transactions opened by the process but not yet validated or canceled, and frees the current selection and current record.

▶ To abort a process:

1 **Select the process on the Process page of the Runtime Explorer.**

2 **Click the Abort button**  **.**

The status of the process in the Runtime Explorer automatically changes to "aborted."

Tracing a Process

You can debug a process by monitoring its execution in the 4D debugger.

▶ To debug a process:

1 **Select the process you want to debug on the Process page of the Runtime Explorer.**

2 **Click the Trace button**  **.**

If the process is being executed, the 4D Debug window appears, allowing you to debug the process by stepping through its execution and evaluating expressions such as the values of fields and variables used in the method.

If the process was paused, 4D "stores" the request and displays the Debug window as soon as the execution of the process resumes. For information about using the Debug window, refer to the *4D Language Reference* manual.

You cannot debug the internal processes created and managed by 4D.

Hiding a Process

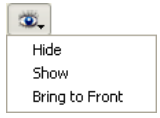
You can make a process invisible in the Application environment by hiding it. When a process is hidden, any windows or menus created by the process are invisible to the user while it is executing.

Hiding a process is useful for operations in which you open a window which you later want to close. Instead of aborting the process to close the window, you can make the window invisible to the user by hiding the process that opened it. Even though the window is hidden, the process continues to execute and complete the operation it began.

► To hide a process:

1 **Select the process you want to hide on the Process page of the Runtime Explorer.**

2 **Click on the display options button and select Hide in the associated menu.**



The process is now hidden from view in the Application environment.

Note The process continues to execute even though it is hidden.

► To display a process again:

1 **Select the hidden process you want show on the Process page of the Runtime Explorer.**

2 **Click on the display options button and select Show in the associated menu.**

The process is displayed again in the Application environment.

Bringing a Process to the Front

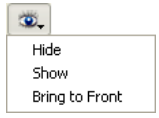
You can make a window the frontmost window by bringing its process to the front. For instance, if the Application process is brought to the front, the Application environment is brought to the front of the screen.

You can bring any user processes to the front. If you have created a window for a process, the window becomes the frontmost window on the screen. If a menu bar is attached to the window, 4D brings the menu bar to the front of the screen and makes its menus the current menus. The current menu bar is replaced by the menu bar of the process that is brought to the front.

► To bring a process to the front:

1 **Select the process you want to bring to the front on the Process page of the Runtime Explorer.**

2 **Click on the display options button and select Bring to Front in the associated menu.**



Any windows attached to the process are brought to the front of the screen. In addition, 4D displays the menu bar for the frontmost process window.

18 Record Management

This chapter reviews the main concepts and functionalities related to managing and working with records in 4D databases. The principles mentioned generally apply to records displayed in Design mode as well as in Application mode. Note that in Application mode, by definition, the interfaces are customized. Consequently, basic record management functions may differ. For example, in Application mode, a record may be added by dragging and dropping it from a hierarchical list — and not via a menu command. However, for each basic function, this chapter indicates the routines or standard actions usually used in the custom interfaces of the Application environment.

The following subjects will be covered:

- Basic concepts concerning record management in 4D, such as the current selection, tables and forms.
- Selections and searches among records.
- Basic record management operations: creating, modifying and deleting records.
- Sorting records.

Browsing Different Tables and Forms

In databases, information is stored in tables. Each table deals with a particular type of information. For example, a “sales contacts” database may include a table that stores data concerning individuals and another table storing data concerning companies.

You use forms to enter and work with your data. Each table of your database has a current input and output form. An input form displays one record at a time. The input form can be used to enter, display and modify the information of a single record. An output form displays several records in a list. Output forms can be used to move between different records, to highlight several of the records and print this selection. It is also possible to enter and modify records directly in an output form.

You can change from one table to another at any time as well as change the input and output forms you are working with in the record display window.

In the Application environment, these changes are usually handled by the custom interface. Moving between different tables and forms is carried out via 4D language commands

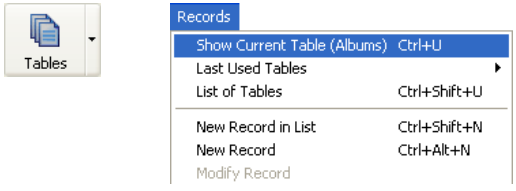
In the Design environment, you have specific commands available. You can change tables and forms at any time.

Each table has a current input and output form. These default forms are specified in the Explorer, for each table of the database. They will be used systematically, unless you call on others during the session.

To choose the tables whose data you want to display, you can use the **Tables** button in the 4D tool bar.
To choose the tables and forms to be displayed, you must use the **List of tables** window. This window can be used to pass quickly from one table to another or from one form to another. You can choose a new table or a new form even when you are already using an input and output form. Your choices are taken into account immediately.

► To move the record display window to the foreground:

1 Click the Tables button in the 4D tool bar.
OR
Choose Show Current Table (*TableName*) in the Records menu.

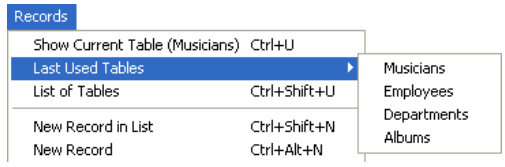


The data of the current table are then displayed in the current output form of the table.

► To change the table of records displayed:

1 Choose a table from the list associated with the Tables button.
OR
Choose a table from the Last Used Tables submenu of the Records menu.

This list contains the names of the last tables displayed during the session:



The data of the table selected are then displayed in the current output form of the table.

► To choose a table or form using the List of tables window:

1 Choose the List of Tables command in the Records menu.

The List of tables window is brought to the foreground.

If necessary, use the scroll bar or resize the window to display the names of all the tables.

2 **Click on the name of a table in the window.**

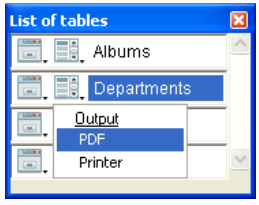
4D displays the current selection of this table in its output form.

Each table name in the window is preceded by two icons symbolizing the input and output forms. The left-most icon lists the Input forms and other icon lists the Output forms.

► To modify the current forms:

1 **Click on the input or output icon of the form table that you want to modify and hold down the mouse button.**

A drop-down menu appears, which lists all the Input or Output forms (depending on the icon you have clicked) of the selected table. The name of the input or output form currently used is underlined in the list.



2 **Select the form that you want to use as the default Input or Output form.**

These settings will remain in effect until you set new ones or exit the database.

Note Default Input and Output forms are defined via the Explorer. For more information, please refer to the paragraph “Designating the Input and Output Forms” on page 368.

Selecting and Searching Records

Selecting records in your database lets you specify information that you want to use. For example, you can search a contacts database to find the telephone number of someone you need to call or select customers to whom you want to send information about your products.

Current Selection

When managing data, you select the group of records with which you want to work. This group of records is called the *current selection*. The current selection can contain none, one, some, or all of the records in a table. Every table has a current selection of records.

The current selection is an important concept in 4D. The most common data management operations are performed on the records in the current selection.

These operations include:

- Sorting records
- Viewing and modifying individual records,
- Updating a group of records,
- Printing a report,
- Generating labels,
- Graphing data,
- Exporting records.

In other words, creating a current selection in a table is the first step towards numerous other data management operations.

The current selection of records is always the set of records most recently selected. For instance, you might have a company database that uses an Employees table to keep track of employee records. Suppose that you decide to search for the records of all engineers in the company.

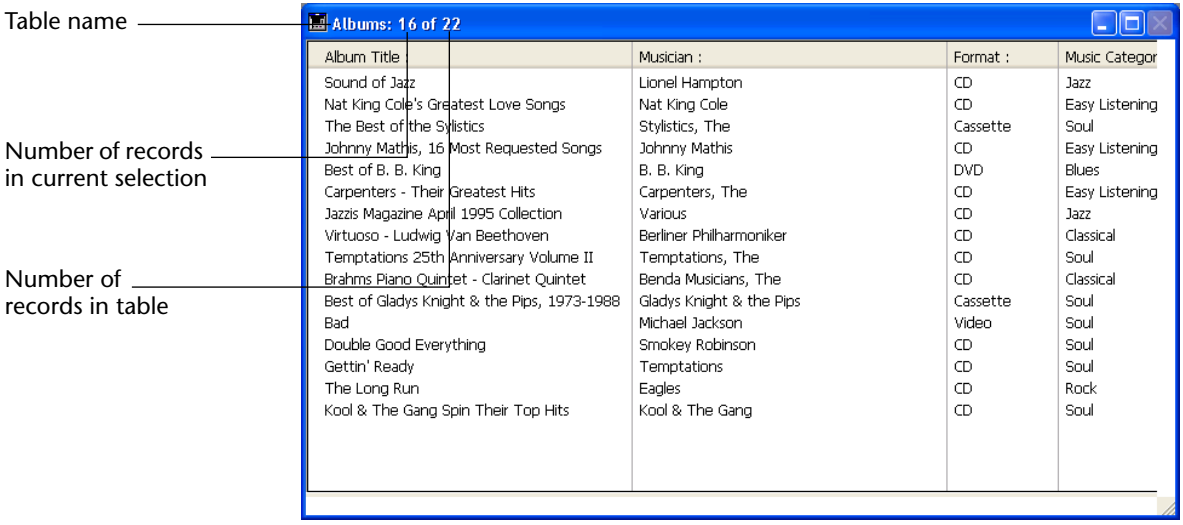
When this query begins, the current selection may contain the records of all employees in the company — salespeople, production personnel, engineers, and so on. When the query is completed, the current selection contains *only the engineers' records*. If you were to print a list of records, the list would contain only the records in the current selection — in this case, the records for all of the engineers in the company. If you were to graph employee salaries, your graph would display the salaries for all of the engineers in the company.

The current selection remains the same until you perform an operation that changes it. You can change the current selection by:

- Selecting all records
- Selecting a subset of records
- Searching for records.

The title bar of the output form tells you how many records are in the current table and how many records from the table are in the current selection.

4D Server With 4D Client, only the number of records in the current selection is displayed.



The control panel of certain input forms displays, under the navigation buttons, the number of the record selected and the total number of records in the current selection. The number of the record selected corresponds to its position in the current selection.

Every table in a database has its own current selection. In a relational database, changing the current selection in one table can change the current selections in related tables. For example, in a database consisting of related [Employees] and [Departments] tables, a opening an input form in the [Departments] table changes the current selection in the [Employees] table. That is, the employees belonging to that department become the new current selection in the [Employees] table. For more information about relations paragraph “Relating Tables” on page 295.

If you use processes for carrying out tasks in the database, there may be several simultaneous current selections per table. Each process acts like an individual 4D environment, which lets you carry out separate tasks. It can be useful to have more than one current selection, particularly when you are comparing two or more types of data, such as the monthly invoicing of several different sales regions. For more information about processes, please refer to the chapter “Processes” on page 929.

Showing All the Records

When using an output form to display records, you can reset the current selection so that it contains all the records in the current table.

In the Application environment, this operation could be carried out via the “Select All” standard action or the ALL RECORDS command of the 4D language.

In the Design environment, you can use the **Show All** command from the **Records** menu. All the records of the current table are then included in the current selection.

Note The **Show All** command is disabled when you are using an input form.

Manually Creating a Subset of Records

You can specify a new current selection in an output form by manually highlighting certain records, then defining them as the new current selection. This is referred to as creating a *subset*.

In the Application environment, you manage records *marked* by users via commands for handling sets and commands of the “Selection” theme.

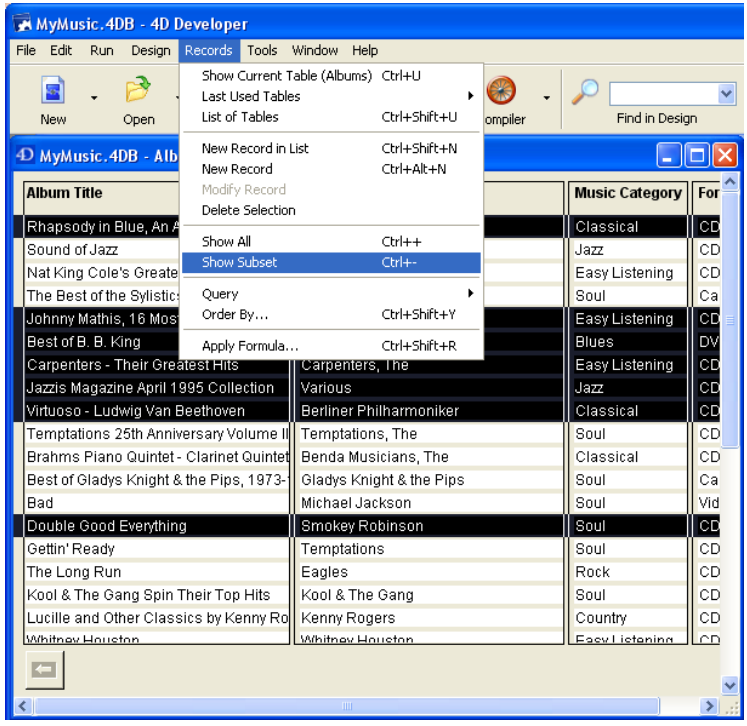
In the Design environment, there is a specific subset command.

► To create a subset in the Design environment:

1 Select the record(s) you want to include in the subset.

The different ways of manually selecting records are described in the paragraph “Highlighting” on page 640.

2 **Choose Show Subset from the Records menu:**



4D displays the new current selection in the output form.

Principles for Searching the Database

Searching is one of the most common database operations. It is often the most convenient way to select the records with which you want to work.

The term *searching* refers to finding a group of records in the database based on the contents of one or more fields. You perform a search by specifying a query. A *query* is the set of instructions that tells 4D which records to include in the new current selection, such as “Company Name is equal to 4D.”

A query always has three elements: *field name*, *comparison operator*, and *value*. The field name is from the current table or a related table. The comparison operator tells 4D how to compare the contents of the field to the value you specify (equal to, greater than, less than, and so forth).

The value specifies the number, string, or other value to which each record is compared.

Suppose you want to see all the records for employees with salaries greater than $30,000. The query you would use is “Salary is greater than 30000.” “Salary” is the field, “is greater than” is the comparison operator, and “30000” is the value.

When you search a database, 4D compares the contents of the field in the query to the value you specify. The new current selection is made up of records that satisfy the rules stated in the query. The new current selection can be no records, one record, a group of records, or all the records in the table.

You can perform a query while you are using either an input or an output form. If you perform a query while you are using an input form, the first record in the new current selection is shown in the input form. You can view, modify, or print the record.

If the new current selection consists of more than one record, you can move through the records using the navigation buttons (Previous Record, Next Record, First Record, Last Record). If you modify a record before pressing a navigation button, 4D will save the modifications to disk.

In relational databases, you can search in fields from other tables, provided that a relation between the tables has been established. For more information about this point, see the paragraph “Relating Tables” on page 295.

If you do the query while using an output form, the new current selection is displayed in the output form. You can reset the current selection to all the records in the current table by choosing **Show All** from the **Records** menu.

Note If a field in the database structure is not used in the current database, the database designer can elect to hide the field by giving it the Invisible attribute. Only tables and fields which are visible appear in the Query editor. For more information about this property, please refer to the paragraph “Setting Table Attributes” on page 261 and to the paragraph “Setting Field Properties” on page 277.

Indexed and Sequential Queries

4D can carry out queries very quickly if it has an ordered list of records to work from. An ordered list is called an *index*. An index is associated with a particular field and is stored on disk as part of the data file.

A query that is performed without an index is slower than an indexed query because the program must start at the beginning of the table and examine each record until it finds the records that meet the criteria you have set. To be sure that it has found all the records you are looking for, it must examine every record in the table sequentially. This process is called a *sequential* search.

If an index is available, the program "knows" where the target records are located. If you are searching for everyone whose last name is Smith, the program will know where in the table the Smiths are located. Thus, it doesn't need to examine every record in the table.

A good analogy for an index is a card catalog in a library. The card catalog is an alphabetized list of all the books in the library. Each record in the catalog contains information about the book's location. If you are looking for a particular book, it would be inefficient to conduct a sequential search of the library's entire holdings. It is much faster to consult the card catalog, obtain the location of the book, and then search the particular shelf on which the book is stored.

When you are designing a database, you should index the fields that you are likely to use often for searching. As you enter or import records, 4D automatically updates all indexes. When you do queries, 4D automatically uses indexes if they are available.

For more information about different types of indexes, please refer to the paragraph "Management of Indexes" on page 288.

Comparison Operators

When you write a query, you tell 4D how to compare the value you specify to the contents of the database. For example, the query, "Last Name equals 'Smith'" uses the "is equal to" comparison operator. It tells 4D to compare the values in the Last Name field to the string "Smith."

Comparisons involving alphanumeric values are not case-sensitive. A search on the last name "Smith" will find records containing "smith," "SMITH," "sMith," and so on.

The following comparison operators are available:

- Is equal to - Is not equal to
- Is greater than - Is greater than or equal to
- Is less than - Is less than or equal to
- Contains - Does not contain.
- Contains Keyword.

Notes • Queries using the Contains and Does Not Contain operators are always sequential queries.
• The Contains Keyword operator is available for fields of the Alpha and Text type only. For more information about this type of query, please refer to the "Comparison Operators" section of the 4D *Language Reference* manual.

Wildcard Character (@)

To make queries easier to specify, 4D has a wildcard character (@) that can replace one or more characters in a search involving an Alpha or Text field. For example, if you are looking for all occurrences of the name "Belmondo" in a field, you may specify the search value in several ways:

A search for:	Finds
Bel@	All values beginning with "Bel"
@do	All values ending with "do"
Bel@do[1]	All values starting with "Bel" and ending with "do"
@elm@	All values containing "elm"

1. In the Preferences dialog box, you can specify that 4D evaluates the @ character (the "at" sign) as a literal character in searches or character string comparisons when it is found within a word (but *not* at the beginning or the end of a word). This option is particularly useful for databases that store e-Mail addresses (which are usually "name@provider.com"). For more information, please refer to the paragraph "Data Management Page" on page 193.

Note It is possible to combine the wildcard with "Contains Keyword" type query only when it is placed at the end of the word being searched for. For example, the search condition "*Notes contains keywords 'anti@'*" is completely valid.

Simple and Compound Queries

You can search on one or more fields. A query on one field is called a *simple* query. For example, the search "Last name equals 'Smith'" is a simple query. When you do a simple query, 4D examines the contents of one field when searching the database.

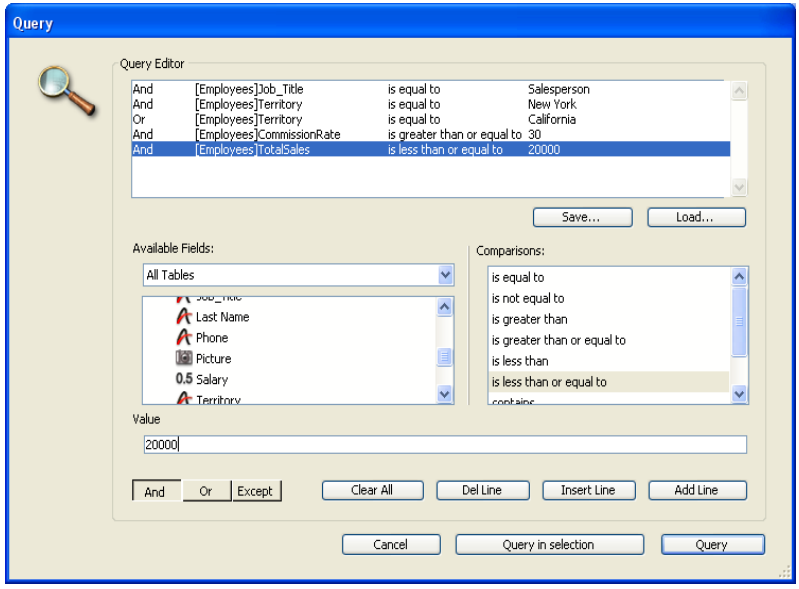
A query on two or more fields is called a *compound* query. When you do a compound query, you combine separate queries using a *conjunction operator.* The conjunction operator tells 4D how to combine the results of the individual queries.

There are three conjunction operators:

- **And**: This operator finds all the records that meet the two conditions simultaneously. For example, the query "Find all the employees who work in the engineering department *and* who make over $50,000" will find the records of only those engineering employees who make over $50,000.
- **Or**: This operator finds all the records that meet either of two simple queries. For example, the query "Find all the employees who work in the engineering department *or* who make over $50,000" will find the records of all the people in the engineering department, as well as all the people who make over $50,000 regardless of the department in which they work.
- **Except**: This operator is the equivalent of "not." The query "Find all the engineers *except* those who make over $50,000" will exclude the engineers making more than $50,000.

The conjunction operators let you create compound queries such as "Find the salespeople in New York *or* California *and* who have commission rates in excess of 30 per cent *and* who had sales volume less than $20,000."

The figure below shows this query being specified in the Query editor:



When this query is executed, 4D finds all the New York and California salespeople who may be getting high commissions for low volume sales. Additional examples of the uses of comparison and conjunction operators are given for each search method.

Ways of Searching

4D provides several powerful editors for searching a database. You can use any of the search tools to create a query. Records that meet the condition become the new current selection.

You can choose any search method when you are using either an input or output form.

In the Design environment, the **Records** > **Query** menu as well as the menu associated with the **Query** button of the 4D tool bar provides four menu commands related to searching. Each menu command displays a different dialog box or window or provides different options. They differ in the types of queries they carry out and the way in which the current selection is displayed.

In the Application environment, the windows of these dialog boxes are available via language commands found in the “Queries” theme.

These four menu commands represent four different search methods:

- **Query...** (QUERY and QUERY SELECTION language commands) : displays the Query editor of 4D. It is a general-purpose search dialog box that can be used to perform simple or compound queries. You can specify compound queries using the conjunctions. You can also save queries to disk and restrict the query to the current selection.
- **Query by Example...** (QUERY BY EXEMPLE language command): displays the current input form for use as a search window. You specify a query by typing the values for which you want to search in the areas corresponding to the fields to be searched. You can specify compound queries by typing values into more than one area. The results of your query are displayed in the current output form.
- **Query and Modify...**: identical to the **Query by Example...** menu command, the difference being that the first record of the selection from the query is loaded, ready to be modified. You can make changes and then browse among found records in order to modify them one by one.
- **Query by Formula...** (QUERY BY FORMULA and QUERY SELECTION BY FORMULA language commands): displays the Formula editor of 4D. You use the Formula editor to construct a query that uses a formula as the query. For example, you can use Query by Formula to search on the last three digits of a six-digit part number. A valid formula returns a Boolean expression (TRUE or FALSE).

Standard Searches

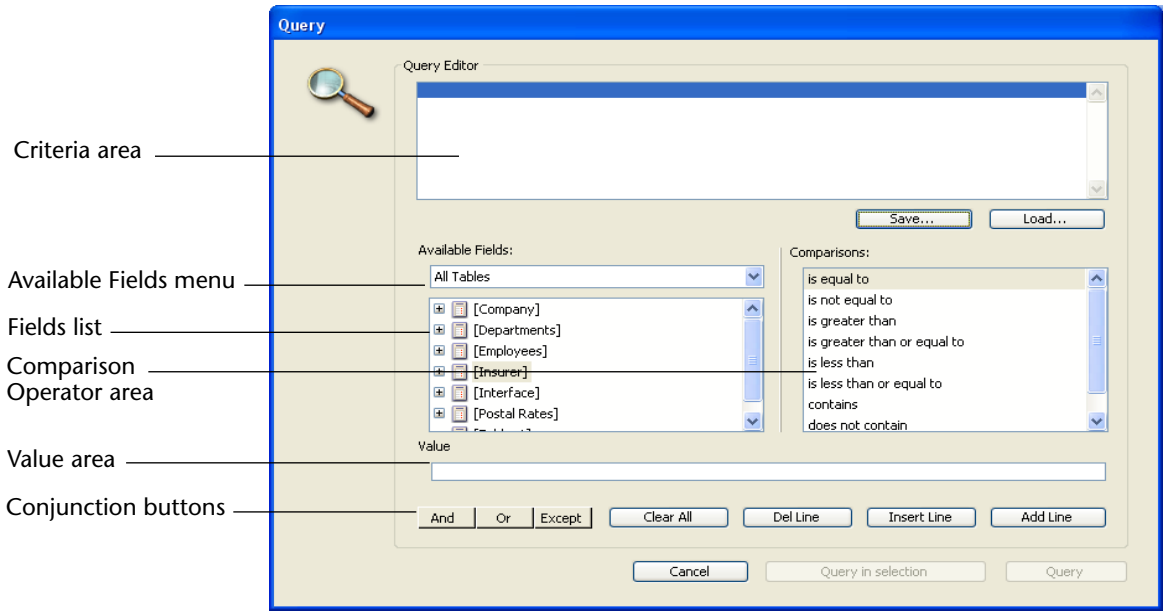
The standard Query editor is a general-purpose editor that can be used to create simple or compound queries. You can create compound searches linked with the And, Or, or Except conjunctions. For example, you can use the Query editor to perform a query for all employees who are over 60 years old *or* who have an income in excess of $45,000.

You have the choice of searching through the current selection of records or all the records in the table. The other three search methods always search the entire table.

You can save queries to disk and open them when you want to repeat the query. The Query editor remembers your last query. You can edit the query or clear it and enter a new query.

You can search in fields of the current table as well as fields of related tables.

The Query editor is shown below:



The Query editor contains the following areas:

- **Criteria area**: This area displays the query as you create it or after you load it from a disk file.
- **Available Fields menu**: This menu allows you to select the table or tables from which you want to display fields in the Fields list. You can display fields from the Master table, the Related tables, or All tables.
- **Fields list**: This area displays a hierarchical list of the fields in the selected table or tables. Indexed fields are shown in boldface.
- **Comparison Operator area:** This area displays a list of comparison operators.
- **Value area**: You enter the value for which you want to search in this area.
- **Conjunction buttons**: This area contains three buttons that correspond to conjunction operators you can use to join the current simple query to the previous simple query.
- **Query in selection button**: This button performs the query only on the records in the current selection.

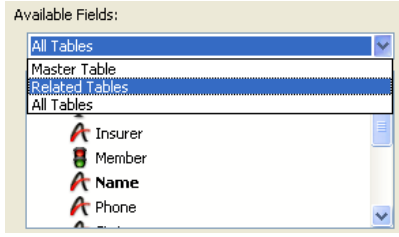
- **Query editor buttons**: You use this area to save your queries, load other queries from disk, cancel the query, or execute the query.

► To create a query condition in the standard editor:

1 **In the Design environment, choose the subcommand Query > Query... from the Records menu, or click on the "Query" button in the tool bar.**
4D displays the Query editor, highlighting the first row of the Criteria area.

Note You can resize the query editor window by clicking on the lower right corner.

2 **Select the table(s) whose fields you want to use for the query.**
You can display the Master Table, the tables related to it or all the tables.



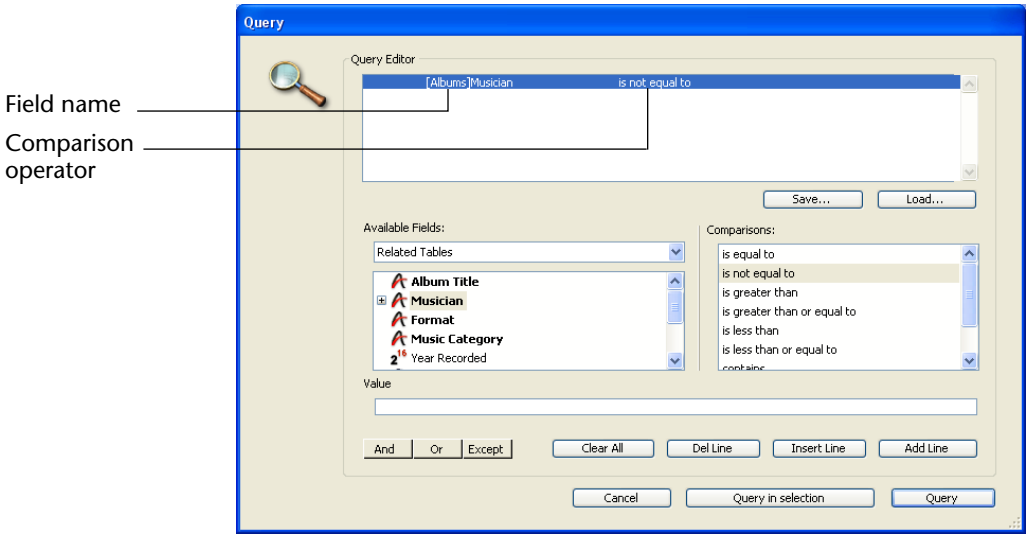
When you select **Related Tables** or **All Tables**, an icon appears to the left of the names of related fields or tables (+ symbol inside a square under Windows, arrow under Mac OS). Clicking on this icon will expand the list of fields for the table.

3 **Click on the name of the field to be used in the search condition.**
4D displays the field name in the criteria area. Its name is preceded by the name of the table to which it belongs.

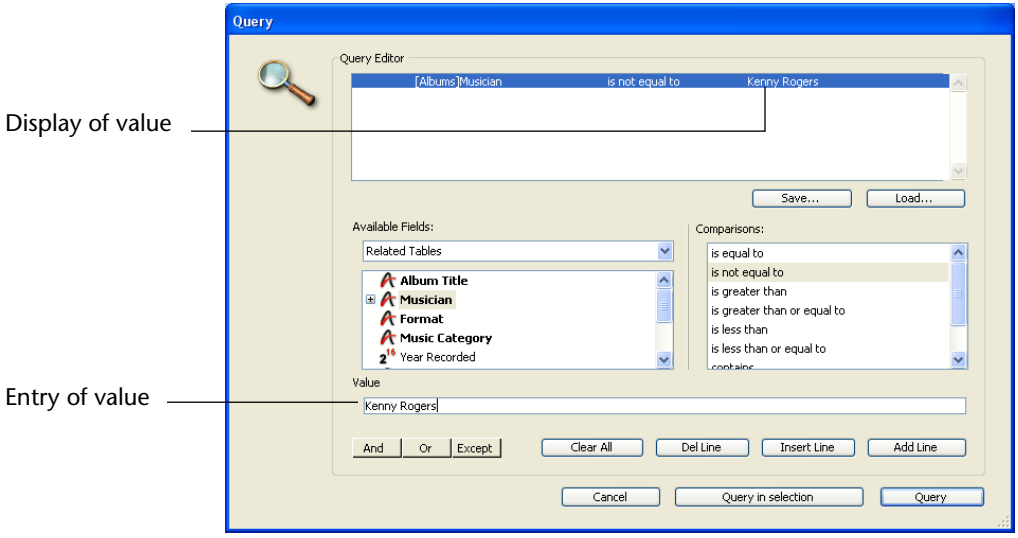
The comparison operators area is highlighted. By default, 4D places the "is equal to" comparison operator after the field name.

4 **If you want to use another comparison operator instead, simply click on the one desired.**

4D places it next to the field name in the criteria area.



5 Type the value you want to search for in the Value area.



In a Text or Alpha field you can use the wildcard character (@) at the end of the value to request a “Begins with” search.

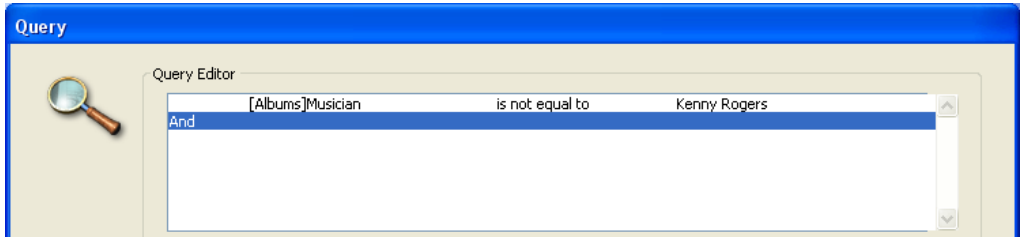
If the field you selected is associated with a choice list, 4D displays the list and prompts you to select a value. If the field you selected is a Boolean field, 4D displays a pair of radio buttons.

6 If you want to add another simple query, click Add Line.

4D adds a new line using the "And" conjunction operator.

7 **If you want to use the "Or" or "Except" operator, click the desired conjunction operator button.**

By default, when you add a line, 4D places the **And** operator in the corresponding area.



8 **Repeat steps 2 through 4 to build the second simple query.**

When you build a compound query, 4D evaluates the simple queries in the order in which they appear in the Query editor (i.e., from top to bottom). There is no precedence among the conjunctions. In other words, *And* does not have priority over *Or*. Therefore, if you use more than two simple queries when building the compound query, the order in which you enter the simple queries can affect the results of the query.

As you build the compound query, you can modify existing parts of the query by clicking the line you want to change and clicking a new field or operator, or typing a new value.

You can remove a simple query by selecting the line and clicking **Del Line**. In a compound search condition, you can remove all the lines by clicking **Clear All**.

Note Be careful, deleting lines cannot be undone.

9 **(Optional) To save the query to disk, click the Save button and enter a filename in the create-file dialog box.**

You do not need to save your query to perform the search.

10 **Click Query to search the entire table.**
OR
Click Query in Selection to restrict the query to the current selection.
OR
Click Cancel to exit the query editor.

■ **Saving a Query to Disk**

If you perform the same query often, you may want to save it to disk.

When you save queries to disk, you only need to create them once. In subsequent uses of the Query editor, you can simply load the desired query from disk and click **Query** or **Query in Selection** to perform it.

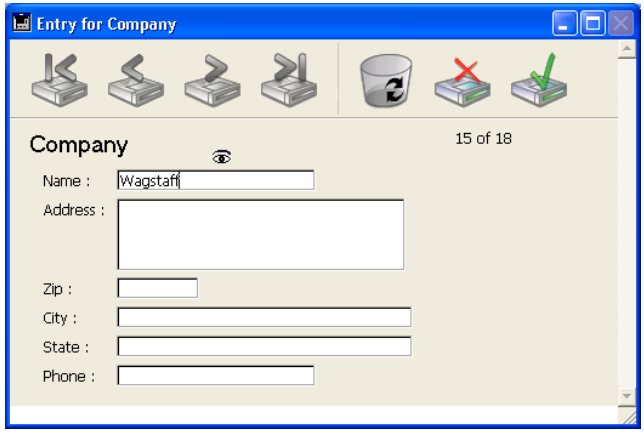
To save a query to disk, click **Save...** in the query editor after specifying your criteria. 4D displays a standard Save file dialog box where you can enter a file name and choose its location on the disk.

Note The file extension for 4D queries is ".4DF".

- **Loading a Saved Query**
  To load a saved query, simply click **Load...** in the Query editor and select the query file (extension ".4DF"). 4D loads your query into the Query editor. When you load a file, it replaces any query that previously appeared in the Query editor.

Query by Example

The Query by Example option is a convenient way to perform many queries. In this type of query, you use the current input form to enter values on which to base the search. You can only search in fields of the current table. Query by Example performs both indexed and sequential queries.

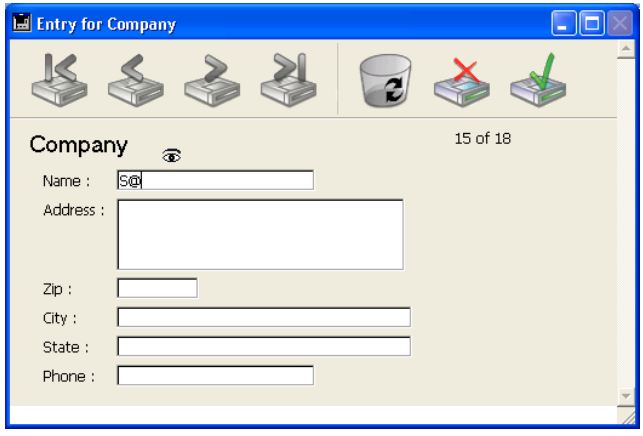


The Query by Example window shown is just an example. Your current input form is always used as the Query by Example window. You can control the appearance of the Query by Example window by changing the current input form using either of the methods described in the paragraph "Browsing Different Tables and Forms" on page 947).

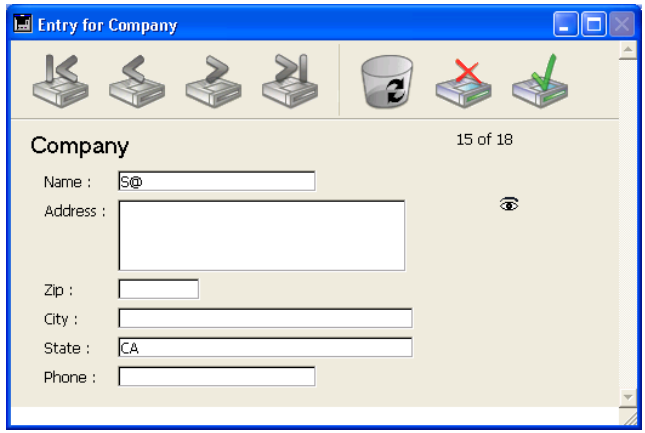
You can use comparison operators in Query by Example. For example, to use the “is equal to” comparison operator, enter the value to be searched for in the appropriate field. If you need a different comparison operator, precede the value with one of the following signs:

Comparison	Operator Sign	Example
is not equal to	#	#Marketing
is greater than	>	>30000
is greater than or equal to	>=	>=30000
is less than	<	<30000
is less than or equal to	<=	<=30000

You can request a “Begins with” query by placing the wildcard character (@) after the value to be searched for. The following figure shows the Query by Example window being used to search for all companies whose names begin with “S”.



You can create a **compound query** by typing values into more than one field. If you enter a compound query, the *And* conjunction is assumed (e.g., "Name starts with "S" *and* is in California"). The following Query by Example window illustrates a compound query:



► To use Query by Example:

1 **In the Design environment, choose Query > Query by Example from the Records menu.**
OR
Choose Query by Example... from the menu associated with the "Query" button in the tool bar.

4D displays the input form for the current table as a Query By Example window. Only the Accept and Cancel buttons are enabled.

2 **Enter a value to search for in a field.**

For example, to find records for everyone with the last name "Smith," you would enter "Smith" in the Last Name field.

To use a comparison operator, precede the value by the comparison operator (see the "Comparison" table on page 966). To do a "begins with" search, follow the value by the "@" symbol.

Note It is not possible to carry out a query using keywords in this editor.

3 **To perform a compound query, enter values in additional fields.**

4D uses the *And* conjunction if you enter values for more than one field.

4 **Click the Accept button or press the Enter key on the numeric keypad to do the query.**

To cancel the query, click the **Cancel** button or use the **Escape** key (or **Esc** key on Mac OS).

The record(s) that meet the search conditions are displayed in the current output form.

Query and Modify

The Query and Modify command is intended for quickly finding and modifying records. Like Query by Example, it uses the current input form as a query window. You specify the query in exactly the same way as with Query by Example.

When a record or selection of records is found, 4D displays the first record in the input form for modification. When you accept the record, you are returned to the output form.

- Using Query and Modify, found records are displayed in the current output form. All records within the new selection are simultaneously displayed.
- Using Query and Modify, the current record of the new selection is also displayed in the current input form. Thus, you can directly replace the values as desired.

  If the new selection contains several records, you can click on the navigation buttons in the import form to scroll through each record and modify the values one by one.

► To use Query and Modify:

1 **In the Design environment, choose Query > Query and Modify from the Records menu.**
OR
Choose Query and Modify... from the menu associated with the "Query" button in the tool bar.
4D displays the current input form as a query window. Only the Accept and Cancel buttons are enabled.

2 **Click the entry area for the first field in which you want to search.**

3 **Enter a value in the criteria entry area.**
For example, to find records for everyone with the last name "Smith," you would enter "Smith" in the Last Name field.

To use a comparison operator, precede the value by the comparison operator. To do a "begins with" query, follow the value by the "@" symbol.

4 **To perform a compound query, enter values in additional fields.**

4D uses the *And* conjunction if you enter values for more than one field. For Alpha fields, you can also use the wildcard character (@) to perform a "Begins with" search. For example, entering "ACME@" in the Company area requests a search of all companies whose name begins with ACME.

5 **Click the Accept button or press the Enter key on the numeric keypad to begin the search.**

4D performs the query, sets the current selection, and displays the first record that meets the search criteria in the input form.

6 **Modify the records using standard text editing methods.**

7 **Click the navigation icons to move to other records in the current selection.**
OR
Click the Accept button or press the Enter key on the numeric keypad to accept the modified record and return to the output form.

4D saves each record as you move to another record.

Click the **Cancel** button to end the editing session — you do not have to modify every record in the current selection.

4D displays the output form.

Query by Formula

Use the Query by Formula editor to find records based on the results of a calculation or data manipulation. You use the Formula editor to write the expression based on which the query will be performed. The expression you write must equal either TRUE or FALSE for each record. You can use any function in the language as well as any project methods specifically designated by the developer.

The Query by Formula command is useful for writing queries that involve operations such as the following:

- operations or evaluations on alphanumeric strings,
- date computations,
- arithmetic computations.

Here are some examples of the uses of the Query by Formula editor.

- The following formula is used to search for records in which the last seven characters of the Phone number field are equal to “2524444.”

  Substring ([Emp]Phone number;4;7) ="2524444"

- The following formula finds people born on the current date regardless of the year:

  (**Day of** (**Current date**) = **Day of** ([Emp]Birthdate)) &
  (**Month of** (**Current date**) = **Month of** ([Emp]Birthdate))

- The following formula divides annual sales by the cost of living and finds records whose calculated value is greater than 1,000:

  ([Stats]Annual Sales / [Stats]CostOfLiving) > 1000

  However, the formula

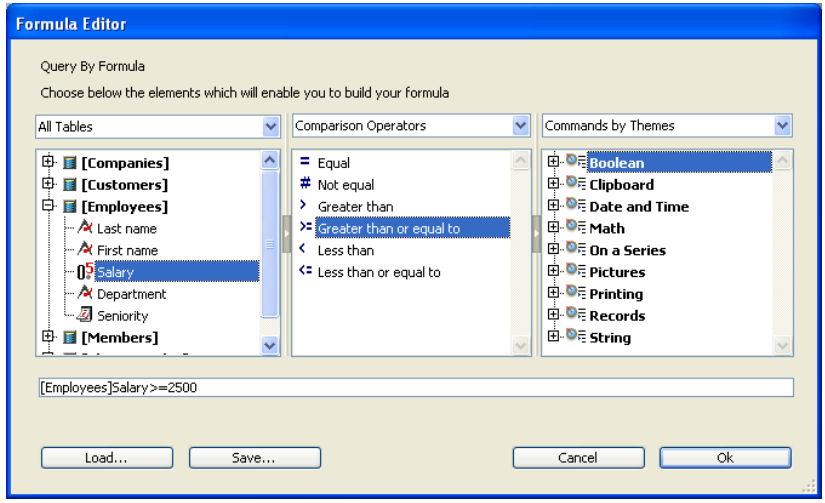
  [Stats]Annual Sales / [Stats]CostOfLiving

  is incorrect because it returns a numeric value, not TRUE or FALSE.

  You can only write formulas that are one logical line long. That is, you cannot press the **Carriage return** key on the keyboard and write a second line. The editing area, however, will wrap to the next line if the statement is too long. If you need to use a formula that is more than one line, write it as a project method and use this method in the formula.

  You can save formulas to disk and load saved formulas into the Formula editor.

The following figure shows a query (search) formula in the Formula editor.



For a detailed description of the Formula editor, see the paragraph "Global Updates" on page 979.

► To use the Query by Formula editor:

1 **In the Design environment, choose Query > Query by Formula from the Records menu.**
OR
Choose Query by Formula... from the menu associated with the "Query" button in the tool bar.

2 **Use the Formula editor to build a formula.**
OR
Click the Load button to retrieve an existing formula from disk.

The expression you write must equal either TRUE or FALSE for each record.

Note You can resize the Formula editor window by clicking on its lower right-hand corner.

If you build a formula in the Formula editor, you can either type your function or use the list of operators and functions.
As you enter values, the editor checks the syntax used. If any syntax errors occur, a message describing the error will appear in the window.

If you click the **Load** button, 4D displays an open-file dialog box where you can select a file. When you load a file, it replaces any formula that appears in the Formula editor.

Note The formula file extension is ".4FR".

3 **(Optional) If you want to save your formula to disk, click the Save... button.**

4D displays a dialog box that can be used to specify the name and location of the file. Click on the **Save** button to save the formula.

Editing Records

This paragraph presents the major operations you perform when working with records in your database. This includes the following:

- Adding records,
- Modifying records,
- Updating records,
- Deleting records.

After you create a database in the Design environment, you can begin to work with your records (entry, modification, deletion, etc.)

These operations are usually carried out in the Application environment via a custom interface.

4D also provides you with the possibility of entering, modifying or deleting records via the Design environment. This means that you can carry out certain basic tasks and test the data of your application.

After adding records to your database, you will usually need to modify the information. If you need to modify one or more records, you can use one of the record selection methods described in the paragraph "Selecting and Searching Records" on page 950 to display them. You can then modify them using either the input or output form. Sometimes you need to make exactly the same change to a group of records in a table. This is called a *global update*. In 4D, you can update the current selection of a table automatically — without having to make the change to each record individually.

You may also find that you need to delete one or more records. You can delete a record if you discover it is outdated or irrelevant. If the record is needed but the values stored in it are no longer correct, you should modify the record rather than deleting it. You can delete the current record from an input form or you can delete a subset of records in the current selection using the output form.

Adding New Records

You can add records using either an input or output form. For more information about the main concepts related to input and output forms, please refer to the paragraph “Input and Output Forms” on page 337.

In the Application environment, records may be added via the “Add Subrecord” standard action (adds a record to a list) or via the ADD RECORD and CREATE RECORD commands.

In the Design environment, 4D provides several adding functions, which are described below.

Input Forms

► To add a new record using an input form:

1 In the Design environment, choose New Record from the Records menu.

You can choose **New Record** while you are using either an input or output form.

4D displays a blank input form and places the insertion point in the first enterable area in the form.

2 **Enter data into the first enterable area then press Tab or the Carriage return to move to the next area.**

Repeat the process of entering data into each enterable area until all data for that record has been entered.

When you have finished entering data and are satisfied with the values you have entered, you can accept the record.

3 **Press the Enter key on the numeric keypad[1], or click the OK button on the input form to accept the record.**

1. Keyboard assignments can be modified in the application Preferences (see the paragraph “Shortcuts Page” on page 171).

By accepting the record, you are requesting that 4D add the record to the database stored on disk.

After you press enter, a new blank input form is displayed.

4 **If you want to create another record, repeat the data entry and validation process.**
OR
When you finish entering data, click the Cancel button on the form or use the Esc key[1] when the next blank record is displayed.
Either of these actions displays the output form.

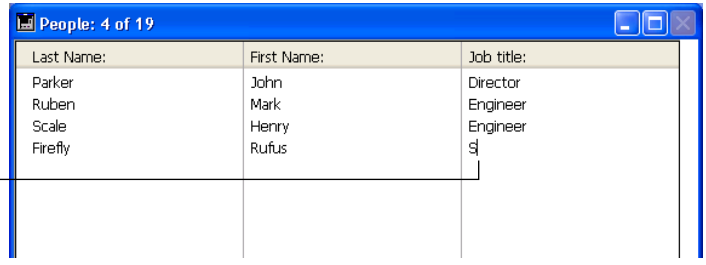
Output Forms

You can also add records directly in the Output form. Keep in mind that in this case you cannot enter data into variables, fields from other tables, or subforms.

▶ To add a new record using an output form:

1 **In the Design environment, choose New Record in List from the Records menu.**
An insertion point appears in the first field below the last record displayed in the output form.

2 **Type in the field and use the Tab or Carriage return key to move through the fields for that record.**



Firefly's job title being entered in List mode

Note Use **Shift+Tab** or **Shift+Carriage return** to move in the reverse direction among the fields of the records.

Any data validation controls that are attached to the fields of the output form will be used when you select those fields. For example, a field with a choice list will display the choice list when you tab into it.

3 **Press the Enter key on the numeric keypad to save the new record and create a new blank record.**
OR
Click a field in another record.
4D accepts the entries in the record you added.

Modifying Records

You modify records when you need to update information or when you discover that the information originally entered is incorrect. Before modifying a group of records, select the records to modify as the current selection. You can search to select records for modification or select the records after highlighting them in the output form. For information about selecting records and searching the contents of a database, refer to the paragraph "Selecting and Searching Records" on page 950.

You can modify records using either the input or output form. The output form is convenient for modifying a group of records because several records are displayed at the same time. However, the output form typically does not include all the fields of the input form and may not duplicate the input form's data entry controls.

If a record is being modified in another process, the record is said to be *locked*. Locked records can be viewed, but they cannot be modified. If you open a locked record, you will be able to view the entries in the fields, but you will not be able to change any data. For a general description of processes, see the chapter "Processes" on page 929.

4D Server In 4D Server, record locking also occurs when a record is being modified by another user.

In the Application environment, records may be modified via the "Edit Subrecord" standard action (modifying a record in list form) or via the MODIFY RECORD command.

In the Design environment, 4D provides several modifying functions, which are described below.

Input Forms

► To modify records using an input form:

1 In the Design environment, highlight a record in the output form and choose Modify Record from the Records menu.
OU
Double-click the record on the output form.
4D displays the record in the default input form.

2 Select certain fields and edit, replace, or delete the values.

3 Click the OK button or press the Enter key[1] on the numeric keypad to accept the modified record and return to the output form.
OR

1. Keyboard assignments can be modified in the application Preferences (see the paragraph “Shortcuts Page” on page 171

Click a navigation button (Previous Record, First Record, Next Record, Last Record) to accept the record and move to another record in the current selection.

Clicking a navigation button moves between records in the current selection.

You can cancel your changes and return to the output form at any time by clicking the **Cancel** button or hitting the **Esc** key.

Output Forms

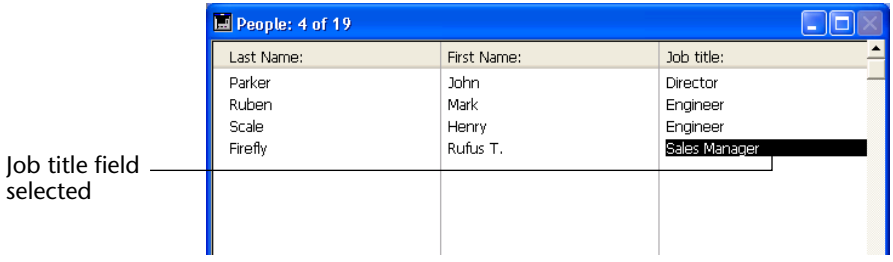
You can modify the fields displayed directly in the output form. Keep in mind that in this case you cannot enter data into variables, fields from other tables, or subforms.

In the Application environment, it is possible to control the possibility of modifying records in list form.

► To modify records using an output form:

1 **Select a record then click on the field to be modified.**

The field in the output form becomes enterable:



Note In the Design environment, you can choose **Modify Record** from the **Records** menu at any time in order to modify the record in the input form.

2 **Type the new text and press Tab or the Carriage return key on the keyboard.**

In the example shown above, typing will replace the job title with a new value. 4D saves your changes to a field when you press **Tab** or **Carriage return** and selects the next field.

3 **Continue modifying fields as needed.**

4 **Click twice on a field in another record of the output form to modify it.**

Global Updates

You do a global update when you want to make a specific change to a group of records. You perform a global update to automate changes to a group of records that would otherwise be tedious and time-consuming. For example, you would perform a global update if you wanted to do the following:

- Change all prices in an Inventory table by a certain percentage,
- Format a numeric or Alpha field.

The global update is done by "applying" a formula to the current selection of records. In other words, the formula is used to make the change to each record in the current selection.

Here are some example formulas and explanations of the functions they perform.

▼ The following formula multiplies the Salary field by 1.05. It could be used, for example, when a salary increase goes into effect.

```
[Emp]Salary := [Emp]Salary * 1.05
```

▼ The following uses a built-in function to make the contents of the State field uppercase. It ensures uniformity in the way State appears in labels and reports.

[Customer]State := **Uppercase** ([Customer]State)

▼ This formula includes a user-written function that formats the Last Name field. It sets the first letter of the Last Name field to uppercase and all the remaining letters to lowercase.

[Emp]Last Name := *Capitalize* ([Emp]Last Name)

The ability to include user-written functions when carrying out global updates is a powerful feature of 4D. For security reasons, access to project methods in the formulas can be restricted.

Formulas can contain 4D language functions as well as project methods (declared "usable" in the forms by the developer).

For more information about programming and the syntax to be applied in your formulas, please refer to the *Language Reference* manual.

You cannot write formulas that are longer than a single logical line, in other words, you cannot hit the **Carriage return** and enter a second line. However, methods that are declared usable in the formula editor can, of course, consist of several lines.

Note The formula editor can also be used for queries (see the paragraph "Query by Formula" on page 969) and sorts (see the paragraph "Sorting Records" on page 987). In these cases, respectively, evaluating the formula for each record lets you find out whether the record corresponds to the search criteria and determines the display order of the record.

Applying a Formula using the Formula Editor

To carry out a global update, you can use the formula editor to write the formula which will then be applied to each record of the current selection.

In the Application environment, you can execute an update formula directly via the EXECUTE FORMULA command or display the formula editor via the EDIT FORMULA command. For more information, please refer to the *Language Reference* manual.

In the Design environment, you have the **Apply Formula...** command in the **Records** menu.

► To perform a global update by applying a formula:

1 **Set the current selection to the records to be updated.**

2 **In the Design environment, choose Apply Formula from the Records menu.**

3 **Use the Formula editor to build the formula.**

The Formula editor is described in the following paragraph.

OR

Click the Load button to retrieve an existing formula from disk.

If you click the **Load** button, 4D displays an open-file dialog box so that you can select a file (".4FR" file extension on Windows). When you load a file, it replaces any formula that currently appears in the Formula editor.

After you load a formula, you can modify it in the editing area.

Note You can resize the Formula editor window by clicking on its lower right corner.

4 **(Optional) If you want to save a new formula to disk, click the Save button.**

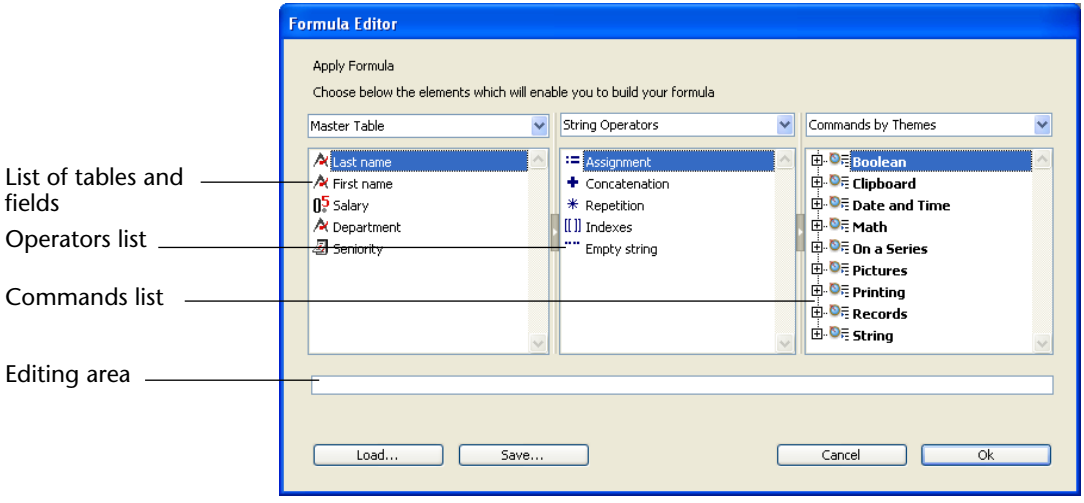
4D displays a create-file dialog box and asks you to name the file in order to store the new formula.

5 **Click OK to apply the formula to the current selection.**

4D closes the editor and applies the formula to the records in the current selection.

Formula Editor

The Formula editor provides many shortcuts for writing formulas. You can click field names, operators and commands, as well as any project methods, to add them to the formula. When you click on an item, it is automatically displayed in the editing area where you can then modify it using standard cut/copy/paste functions. You can also enter items directly in the editing area or drag and drop them from the list of items.

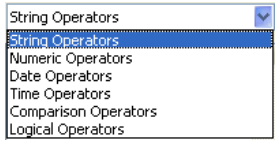


The Formula editor contains the following areas:

- **List of tables and fields**: This area displays the fields of the table. The menu located above the list lets you set the fields to be displayed. You can use fields of the current table, those of related tables or those of all the tables.

Note Tables and fields with the Invisible property do not appear in the list. For more information about this property, please refer to the paragraph “Setting Table Attributes” on page 261 and to the paragraph “Setting Field Properties” on page 277.

- **Operators list**: The operators list lets you choose the operators to be used in the formula. The operators are grouped into themes found in the menu located above the list:



  Each theme displays all the available operators for the corresponding type of data or operation. For example, the assignment operator := is available for all data types.
  For a description of each operator, refer to the following paragraph.

- **Commands list**: The commands list contains the 4D functions that can be used in formulas, as well as any project methods allowed by the developer. The menu located above the list lets you display the commands by theme or by alphabetical order. Refer to the *4D Language Reference* manual for a description of the 4D commands that appear in this menu.

Note In principle, project methods that can be used in formulas must be declared beforehand using the 4D SET ALLOWED METHODS command. However, by default, the Designer and Administrator of the database have complete access to the 4D commands and user methods in the Formula editor. It is also possible to completely disable access control for all users. These options are set via the Preferences of the application (for more information, please refer to the paragraph “Access Page” on page 166).

Formula Operators

Here is a brief description of the different operators available in the Formula editor. For a more detailed description of the possibilities provided by these operators, refer to the “Operators” chapter of the *4D Language Reference* manual.

■ **String Operators**
A and B are character strings; N is a number.

Operator	Use	Description
:= Assignment	A:=B	Assigns the value B to A
+ Concatenation	A+B	Returns AB
* Repetition	A*N	Repeats the value of A N times
[[ ]] Indexes	[[A]]N	Returns the Nth character of A
"" Empty string	""	Inserts a pair of quotation marks

■ **Numeric Operators**
X and Y are numbers.

Operator	Use	Description
:= Assignment	X:=Y	Assigns the value Y to X
+ Addition	X+Y	Returns X plus Y
- Subtraction	X-Y	Returns X minus Y
* Multiplication	X*Y	Returns X multiplied by Y
/ Division	X/Y	Returns X divided by Y
\ Integer Division	X\Y	Returns the integer division of X by Y (X and Y must be integers)
% Modulo	X%Y	Divides X by Y and returns the remainder
^ Exponentiation	X^Y	Returns X to the power of Y

Note The modulo % operator returns significant values with numbers belonging to the long integer category (from -2^31 to +2^31 minus 1). To calculate the modulo of numbers outside of this interval, use the Mod command.

■ **Date Operators**
D1 and D2 are dates; N is a number.

Operator	Use	Description
:= Assignment	D1:=D2	Assigns the value D2 to D1
+ Addition	D1+N	Returns D1 plus N days
- Difference	D1-D2 or D1-N	Returns the number of days between D1 and D2 Returns D1 minus N days
!//! Blank date	!00/00/00!	Inserts a blank date

- **Time Operators**
  H1 and H2 are times; N is a number.

Operator	Use	Description
:= Assignment	H1:=H2	Assigns the value H2 to H1
+ Addition	H1+H2 or H1+N	Returns H1 plus H2 Returns H1 plus N seconds, expressed in seconds elapsed since midnight
- Subtraction	H1-H2 or H1-N	Returns H1 minus H2 Returns H1 minus N seconds, expressed in seconds elapsed since midnight
* Multiplication	H1*N	Returns H1 multiplied by N, expressed in seconds elapsed since midnight
/ Division	H1/N	Returns H1 divided by N, expressed in seconds elapsed since midnight
\ Integer Division	H1\N	Returns the integer division of H1 by N, expressed in seconds elapsed since midnight
% Modulo	H1%N	Divides H1 by N and returns the remainder
?::? Blank hour	?00:00:00?	Inserts a blank hour

- **Comparison Operators**
  Z1 and Z2 can be of the string, numeric, date or time type.

Operator	Use	Description
= Equal	Z1=Z2	Returns True if Z1 equals Z2
# Not equal	Z1#Z2	Returns True if Z1 does not equal Z2
> Greater than	Z1>D	Returns True if Z1 is greater than Z2
>= Greater than or equal to	Z1>=Z2	Returns True if Z1 is greater than or equal to Z2
< Less than	Z1<Z2	Returns True if Z1 is less than Z2
<= Less than or equal to	Z1<=Z2	Returns True if Z1 is less than or equal to Z2

- **Logical Operators**
  B1 and B2 must be Booleans (expressions that are TRUE or FALSE).

Operator	Use	Description
& AND	B1 & B2	Returns True if B1 is True and B2 is True
| OR	B1 | B2	Returns True if B1 is True or B2 is True

Deleting Records

You may want to delete a record that is outdated or no longer necessary. If the record is needed but the values stored in the record are incorrect, you should modify the record rather than delete it.

You can delete records in two ways:

- Delete a record individually (usually from an input form).
- Delete a set of records (usually from an output form).

The deletion of records is carried out via the "Delete Record" or "Delete Subrecord" (deletion in list) standard actions or via the DELETE RECORD or DELETE SELECTION commands.

In the Design environment, you can also use the **Clear** command of the **Edit** menu as well as the deletion keys.

Deleting records is permanent and can only be undone by restoring a database backup. When you delete records, 4D displays a dialog box asking you to confirm the deletion.

Input Form

Deleting records from the input form lets you verify the contents of each record before you delete it.

▶ To delete records using the input form:

1 **Open the record you want to delete.**

2 **Click the Delete button to delete the record.**

Depending on the design of the input form, the **Delete** (or **Clear**) button may be represented in different ways. By default, it is represented by an icon symbolizing a trash can which is associated with the "Delete record" standard action. It is also possible for a form to not have a delete button.

4D asks you to confirm the deletion. You cannot undo the deletion after 4D removes the record.

3 **Click the Yes button to complete the deletion.**

4D removes the current record from the database and returns to the output form.

Output Form

Using the output form, you can delete several records in one operation. The records to be deleted must be highlighted in the output form.

► To delete records using the output form:

1 **Highlight the record or records that you want to delete.**

Highlighting records is described in the paragraph “Highlighting” on page 640.

2 **In the Design environment, choose Clear from the Edit menu or press the Delete or Backspace key.**

4D displays a dialog box asking you to confirm the deletion. You cannot undo a deletion.

3 **Click OK to complete the deletion.**

4D removes the highlighted record or records from the database.

Note To delete all the records of a table, choose **Show All** from the **Records** menu then **Select All** from the **Edit** menu before choosing **Clear** from the **Edit** menu (or using a deletion key).

Deleting Locked Records You cannot delete locked records. Records are locked when they are being used by another process. For instance, if a process opens a record for modification, 4D locks that record so that other processes cannot modify it.

4D Server Records are also locked when they are being used by another user.

Before deleting records, you create a selection of the records you want to delete. If your selection includes any locked records, the deletion will proceed but the locked records will not be deleted and will remain in the current selection after the deletion. You must wait until these records are unlocked (i.e., no longer being used) to delete them. The commands of the "Record Locking" theme can be used to manage this type of scenario.

Records Deleted in Another Process

The current selection may be altered by records being deleted in another process. For example, while you are working in your database, you might start another process that deletes certain records from a table. The records deleted in that process are permanently removed from the table. However, the records you see while working with the database may not reflect those changes to the table until a new selection of records is created.

To illustrate this point, suppose that a table contains fifty records and that all of the records are in the current selection. At this point, the title bar of the output form says that "50 of 50" records are selected. If one of the records is deleted in another process, the title bar changes to say that "50 of 49" records are selected. There now appears to be more records in the current selection than in the table! The title bar will be updated when you change your current selection.

If you attempt to modify or delete the deleted record, a dialog box will appear saying that the record has been deleted.

4D Server Records deleted by another user have the same effect on the current selection. The records are deleted from the table, but not from the current selection. Thus, the current selection may appear to contain more records than actually exist in the table.

Sorting Records

A *sort* reorders records according to the values in the table. It is common to sort records:

- To view records on screen in a particular order
- Before printing a report or labels
- Before graphing data.

As you enter data into a new database, 4D stores the records in the order in which they were entered or imported. When you list records in an output form or print records, they appear in this order. Often, you want to view records in another order. For example, you might want to alphabetize a list of names in a report — a sort on the Last Name field reorders the records alphabetically by Last Name.

A sort can be done while you are using an input or output form. If you sort from an input form, the first record in the new sort order is displayed in the input form. Otherwise, the sorted records are displayed in the output form.

4D conducts indexed sorts very quickly. If you are sorting only one field and that field is indexed, 4D uses the index.

Sorting Via the Order By Editor

To carry out a sort, you can use the Order By editor, which lets you manually set the sort criteria.

In the Application environment, you can execute a sort directly via the ORDER BY or ORDER BY FORMULA command. However, it is possible to display the Order By editor via these commands. For more information, please refer to the *Language Reference* manual.

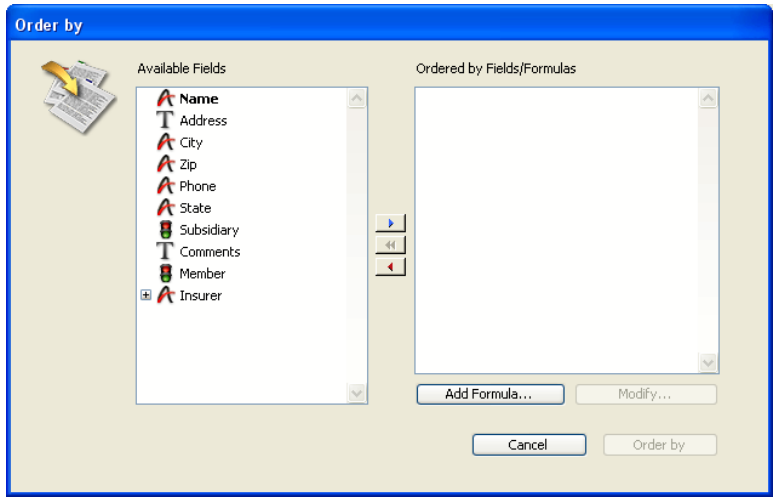
In the Design environment, you have the **Order By...** command in the **Records** menu.

Sorting reorders the current selection of records. This sort is temporary and applies only to the current selection; it does not affect the order in which records are stored in the database. A sort is commonly used immediately after a query and before printing a report or labels.

▶ To sort the current selection:

1 **In the Design environment, choose Order by from the Records menu or from the menu associated with the "Query" button in the 4D tool bar.**

4D displays the Order By editor which is described in the following paragraph.



When you open the Order By editor, the Fields list displays the fields from the master table. You can choose fields from related tables by expanding the foreign key field.

2 **Double-click the name of a field which you want to sort.**
OR
Select the field then click on the add button  **found in the central part of the window.**
OR
Drag the field name from the Fields list to the Sort area.

3 **To sort based on a formula, click the Add Formula button**

4D displays the Formula editor. For more information about this point, please refer to the paragraph “Sorting Based on a Formula” on page 993.

The field name (preceded by the name of the table to which it belongs) or formula appears with a sort direction arrow in the Criteria area.

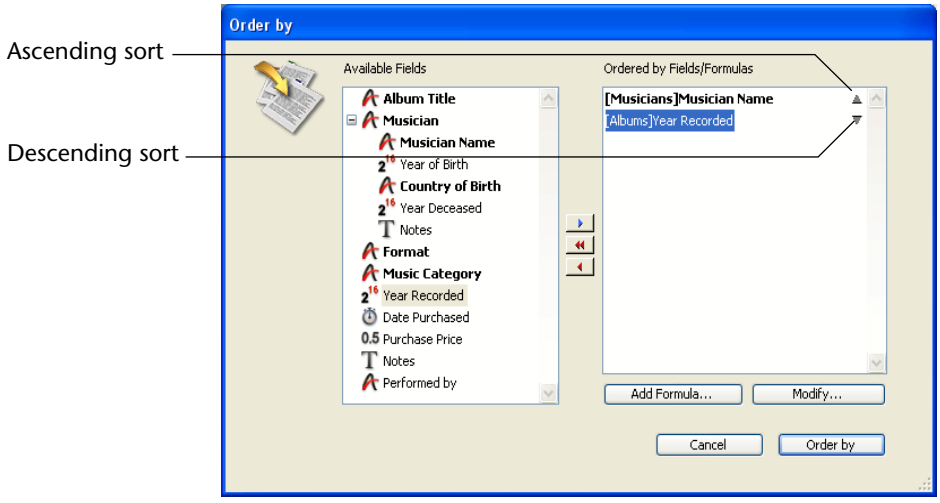
4 **If necessary, click the sort direction arrow in the Sort area to switch between ascending and descending sort order.**

Up is ascending; down is descending. For more information, please refer to the paragraph “Ascending and Descending Sorts” on page 993.

5 **If necessary, repeat steps 2 to 4 to add secondary sort levels to the list.**

For more information about sort levels, please refer to the paragraph “Sort Levels” on page 992.

For each sort level, you can specify ascending or descending order.



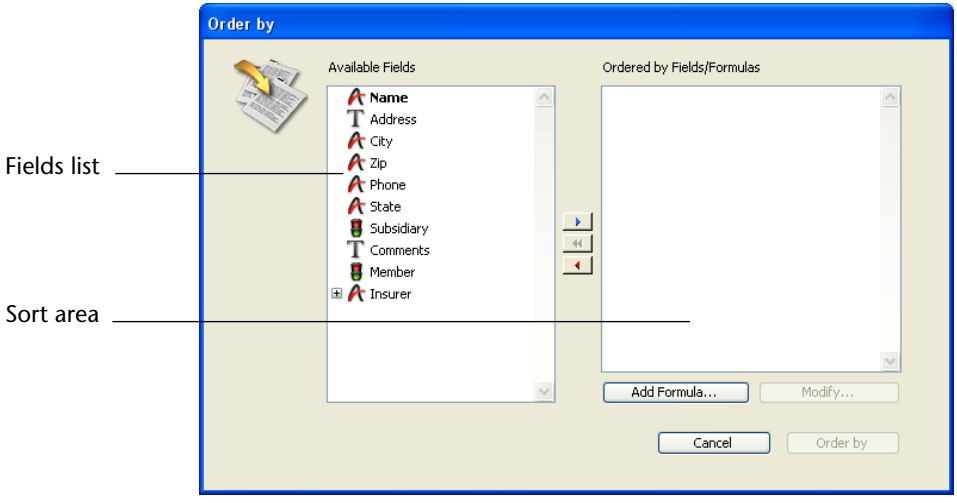
For example, the sort order shown requests a listing of musicians sorted by name. The albums of each musician are sorted by the year they were recorded, starting with the most recent.

6 **Click the Order By button.**

4D sorts the current selection. If you are using an input form, the first record in the sorted current selection is displayed.

The Order By Editor

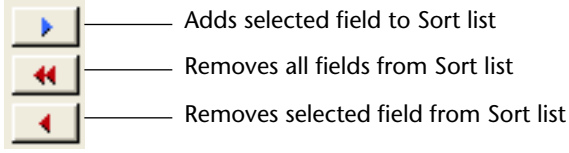
The Order By editor contains the following areas and commands:



- **Fields list**: This area displays a hierarchical list of fields in the current table. Indexed fields are shown in boldface. You can sort on fields from subtables and related tables (provided the relation is automatic). To use a field from a related table or subtable, expand the foreign key field by clicking on the plus sign (on Windows) or arrow (on Macintosh) to display the fields from the related table.

Note Tables and fields with the Invisible property do not appear in the list. For more information about this property, please refer to the paragraph "Setting Table Attributes" on page 261 and to the paragraph "Setting Field Properties" on page 277.

- **Sort area**: This area displays the sort fields or sort formulas and the direction of each sort. The arrows on the right of this area are used to specify an ascending or descending sort.
- **Button panel**: The Button panel has buttons for adding or removing fields from the Sort list.



- **Add Formula button**: You use the **Add Formula** button to write a formula as one of the sort criteria. You use a formula when you want to sort something that is not a field — such as a calculated value or a portion of a field. For instance, you might sort a calculated failure rate for parts or the last four digits of a Part Number field.
- **Modify button**: When you click the **Modify** button, it displays the selected sort criterion in the Formula editor. If the selected criterion is a formula, the formula is presented for editing. If the criterion is a field, the field name appears in the editing window of the Formula editor.
- **Cancel button**: You use this button to cancel the sort and return to the form you were using.
- **Order By button**: You use this button to perform the sort.

Note The Quick Report editor can sort records that appear in a quick report. For more information about the Quick Report editor, see the chapter "Quick Reports" on page 699.

Sort Levels

You can sort records in up to 30 different fields or formulas. Each field or formula you sort is referred to as a *sort level*. For example, the results of a two-level ascending sort of the last name and first name fields would produce a list such as this:

Aardvark, Anthony
Aardvark, Artemis
Aardvark, Arthur
.
.
Zygote, Elena
Zymosian, Elmer

When sorting the contents of fields, 4D is not case sensitive (i.e., “Smith” = “smith”) or diacritical sensitive (i.e., “Aá” = “Aa”).

Note If you are using an international version of 4D and your operating system is diacritical sensitive (e.g., “Aá” ≠ “Aa”), sorting operations will also take diacritical marks into account.

Reordering Sort Levels

You can decide to modify the order in which the sort levels have been specified or to remove one or more level(s) at any time.

► To reorder sort levels, use drag and drop:

1 **In the sort area, click on the level to be moved and hold down the mouse button.**

2 **Drag the level through the list and drop it where you want to place it.**
The level will be inserted just above the level where it is dropped.

► To remove a sort level:

1 **In the sort area, highlight the level you want to remove.**

2 **Press Backspace.**
OR
Click on the deletion button  **.**
The sort level is removed from the area.

► To remove all the sort levels:

1 **Click on the global deletion button**  **.**
All the sort levels are removed from the area.

Ascending and Descending Sorts

You can specify either an ascending or descending order for each field or formula that you are sorting. Sorting from A to Z or smallest to largest is known as an *ascending sort*. Sorting in the reverse order is called a *descending sort* — largest to smallest, latest to earliest, and Z to A.

If you are sorting more than one level, you can freely mix ascending and descending sort orders. A multiple-level sort can mix fields and formulas, and ascending and descending sort orders.

▶ To modify the sort order of a level:

1 Click on the arrow to the right of the level in the sort area.

Sorting Based on a Formula

You can sort a field or sort based on a formula. For example, the following formula sorts the month of a birth date field:

Month of (Birth Date)

▶ To create a sort formula:

1 In the Order By editor, click the Add Formula button.
4D displays the Formula editor.

You use the Formula editor to create a formula that returns the values to be sorted. The formula can return values of any data type. For more information about how to use the Formula editor, see the paragraph "Formula Editor" on page 981.

2 Click OK when you are finished writing the formula.
4D displays the formula in the Sort area.

The formula appears with a sort direction arrow in the Sort area.

▶ To modify a formula or create a formula using a field already placed in the sort area:

1 Select a formula or a field name in the sort area and click on Modify...
OR
Double-click on a formula or a field name in the sort area.
The Formula editor appears containing the selected formula or field name.

2 Make your changes and click on OK.

Sorting While a Record is Being Modified

If a record in the current selection is being modified in another process at the same time that you are performing a sort, 4D uses the last saved value of the field in the sort.

For example, if an employee's job title is in the process of being changed from "Engineer" to "Staff Engineer" while you are sorting on Job Title, the employee record will still be listed as "Engineer" and sorted with the engineer records.

If you were to perform the sort again, after the record has been modified and saved, the new job title would be used and the employee record would be sorted as a staff engineer.

4D Server The last saved value of a field is also used if a record in the current selection is being modified by another user. If the sort is performed again, after the other user has finished modifying the record, the newly saved value of the field is used.

Sorting While an Index is Being Built

Sorting an indexed field is extremely rapid, since the values of the field have already been organized by 4D. However, if the index for a field is being built or rebuilt at the same time that you are performing a sort on the field, 4D will be unable to use the index and the sort will be sequential.

If you sort an indexed field and do not want to do a sequential sort, check that the index is not being built before launching the sort.

Printing Records

You can print the records of your tables at any time. The records are printed via database forms.

In the Application environment, printing records will usually be handled by buttons or custom menus. The records will be printed via dedicated forms. The commands of the "Printing" theme can be used to set up custom printing processes.

In the Design environment, 4D lets you print one or more records and choose the form to be used. You can print a list of records (a report) or records in page mode.

4D prints the records in the current selection. Set the current selection to the subset of records you want to print. If you want the report to list the records in a particular order, sort the current selection before printing the report. If you want to print a single record in page mode, double-click it to open it.

► To print records:

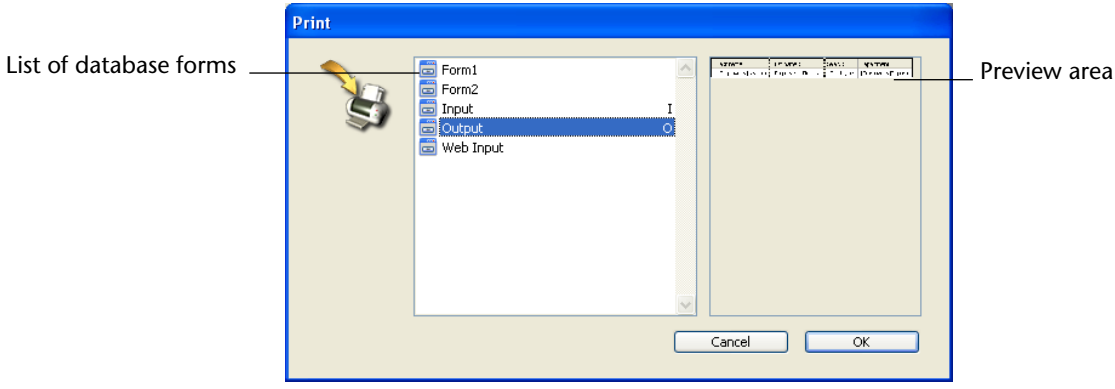
1 **In the record display window, select the records to be printed.**

2 **(Optional) Sort the records.**

This step is mandatory if you are printing records in a report with breaks. In this case, you must sort the records on at least one level more than the number of break levels of the report. For more information about this point, please refer to the paragraph “Reports with Breaks” on page 666.

3 **Choose Print from the File menu.**

The Print Form dialog box appears.



4 **Choose the form you want to use.**

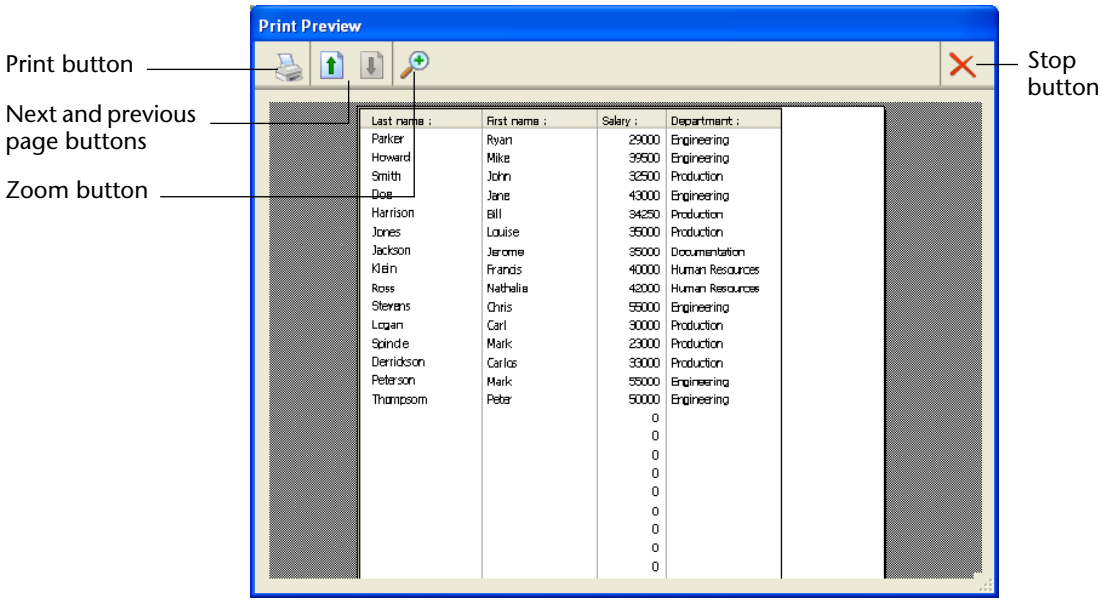
When you click on a form, a preview appears in the preview area.

5 **Click OK.**

4D displays the Page Setup dialog box for your selected printer. Choose any desired options and click **OK**.

4D displays the Print dialog box for the printer you selected in the Print Manager. To preview the report on the screen, check the Preview on Screen check box.

If you checked the **Preview on Screen** check box, the report will be displayed on screen, one page at a time. The figure below shows a report previewed on screen.



When a page of the report is being previewed, you can do the following:

- View a close-up of the report by clicking the **Zoom** button. To exit zoom mode, click again on the zoom button.
- When you use Zoom, you can move around the page by dragging the hand pointer.
- View the next page or previous page by clicking the **Next Page** or **Previous Page** buttons. These buttons are dimmed when the report consists of a single page.
- Cancel printing by clicking the **Stop** button.
- Print the page being previewed by clicking the **Print** button.

If you did not check the Preview on Screen check box, the report will be sent to the printer immediately.

19 Importing and Exporting Data

4D's data importing and exporting capabilities provide a fast and reliable way to transfer information to and from your database. If you are upgrading to 4D from another database or a spreadsheet, you can avoid the work of re-entering the data from the keyboard by importing your data. You can import data from Windows data management applications, or any non-Windows application whose data has been transferred to Windows.

You can also export data from 4D so that it can be used in other types of programs that process information. For example, you can export data to a specialized graphics or statistics program. You can also transfer the exported data to another type of computer.

You can import or export data using the settings in the appropriate dialog boxes or using a form. Importing or exporting using the dialog boxes is faster than using a form. However, if you use a form you can use object and form methods to process data. When you import or export data using a form, the form and object methods are executed for every record that is processed.

When importing or exporting data via a file, you specify the file format, the field and record delimiters, and the fields to be imported or exported. Once you have defined your import or export settings, you can save or load them into or from Import/Export Settings files.

Finally, 4D lets you import and export data from and to an ODBC data source, without passing by an intermediary file. This point is covered in the paragraph "Importing and Exporting Data via an ODBC Data Source" on page 1023.

Note You can also use the Quick Report editor to export records. For more information, please refer to the **File** option in the paragraph "Executing a Quick Report" on page 748.
You can export and import data using the 4D language commands. Refer to the "Import and Export" section of the 4D *Language Reference* manual.

File Formats

The file format describes the way in which fields and records are arranged in the import or export file. Some file formats also include formatting information. 4D supports the following file formats:

- **Text (ASCII)**: This format separates fields within a record with the end-of-field delimiter and records with the end-of-record delimiter. The default field and record delimiters are the Tab and Return, respectively. You can also include field names as the first row of an export Text file. If you include field names as the first row of an import Text file, you can tell 4D to use those names when it imports the file; otherwise, you can instruct 4D to ignore the row of field names.
- **Fixed Length Text Format**: The Fixed Length Text format allocates a specific number of characters per field. No end-of-field delimiter is used. All records have the same number of characters even if a the record's values can fit into fewer characters. When this happens, a fill character — usually a space — is used to pad fields. You can specify the fill character for each data type or use default characters (see the "Filling Page" sections on page 1011 and page 1022). However, if a value has more characters than the number of characters allocated to the field, the value will be truncated.
- **SYLK**: This stands for SYmbolic LinK format.
- **DIFF**: This stands for Data Interchange Format.
- **DBF (dBase)**: dBase is a file format commonly used in DOS and Windows database applications.

- **XML**: XML (eXtensible Markup Language) is a data exchange standard. This language is based on the use of tags that enable precise description of the exchanged data as well as its structure. XML files are Text format files; their content is parsed by the applications importing the data. Many applications support this format.
  For more information about XML terminology, refer to the *4D Language Reference* manual.
- **4D Application**: This format, which is specific to 4D, makes it easy to exchange records between different 4D databases. This format supports all of 4D's data types, including Picture and Blob. The 4D export file extension is ".4IE."

Macintosh and Windows Files

When you import or export data, you can specify whether the import/export file will be for either the **Macintosh** or **Windows** environments. This option affects the choice of field and record delimiters[1] for Text files and whether an ASCII map is used (in non-Unicode mode):

- **Macintosh file**: The delimiters will be the standard ones used on Mac OS (End of field= **Tab**, End of record= **Return**, End of file= <**None**>).
- **Windows file**: The delimiters will be the standard ones used on Windows (End of field = **Tab**, End of record= **Return+Line feed**, End of file = <**None**>).
  Moreover, when the database is running in non-Unicode mode, in this case 4D's Mac OS/Windows conversion filter is used (remember that an internal filter ensures multi-platform compatibility of 4D databases by storing the data in Macintosh ASCII format).

  For example, if you choose **Macintosh file**, the Carriage return (ASCII code 13) will be used as the record delimiter. When importing a Text file, 4D automatically converts any instances of Return plus Line Feed to Return only.

Note ASCII codes in 4D are given in decimal (base 10) numbers. A table of ASCII codes is included in the *4D Language Reference* manual.

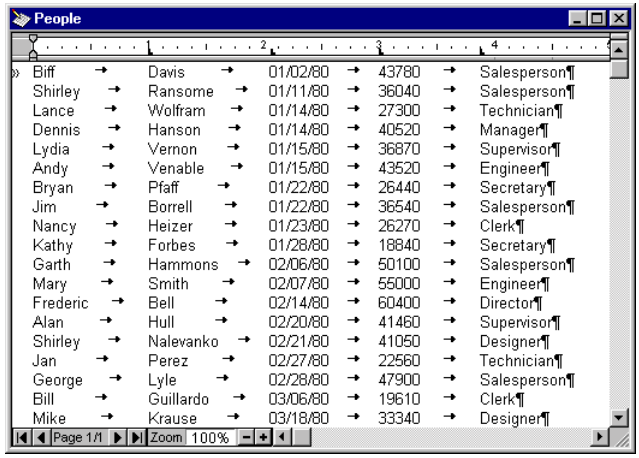
1. This option allows you to pre-enter values in the Delimiters Page of the Import or Export editor. You can also modify them directly on this page.

Field, Record, and File Delimiters

End-of-field delimiters are placed between fields in a record and the end-of-record delimiter is placed after each record. End-of-file delimiters are placed at the end of the import or export file. They may be necessary when exchanging files with certain applications. Delimiters are used only with the Text and Fixed Length Text formats. The Fixed Length Text format does not use Field delimiters and, generally, neither text format uses an explicit File delimiter.

The following figure shows a text file as displayed in a word processor. The file is in Text format. This word processor displays the Tab character as an arrow → and the Carriage return as a paragraph symbol ¶.

This is a Text format file:



When importing data using the Text format, 4D uses the delimiters embedded in the file to determine where fields and records end. When exporting data, 4D automatically places these delimiters in the file for you.

Be certain that fields do not contain embedded characters that are also used as field or record delimiters. For example, if the Carriage return is used as the record delimiter, no field should contain a Carriage return. When importing data, 4D will interpret any Carriage return as a record delimiter. Thus, a Carriage return inside a field will be erroneously interpreted as the end of that record. An improperly placed Carriage return will disrupt the importing process.

When you export data that includes a Text field, a Carriage return might be embedded within the Text field. If this is a possibility, choose another record delimiter or remove the embedded Carriage returns prior to exporting the data.

Importing Data from Files

You can import data from files in XML, 4D, SYLK, DIFF, DBF, Text, or Fixed Length Text formats. If you are importing data that has been exported from another application, see that application's documentation for information about exporting the data in one of these formats. If the other application uses a different format for exporting data, you may need to modify the file in advance using a text editor or word processor.

If you are importing data from another application running on the same platform as your database, first export the data using the other application.

Note the order in which the fields were exported, the format the application uses to save the data, and, if the Text format was used, the delimiters used by the application. When importing the data, you must match these settings.

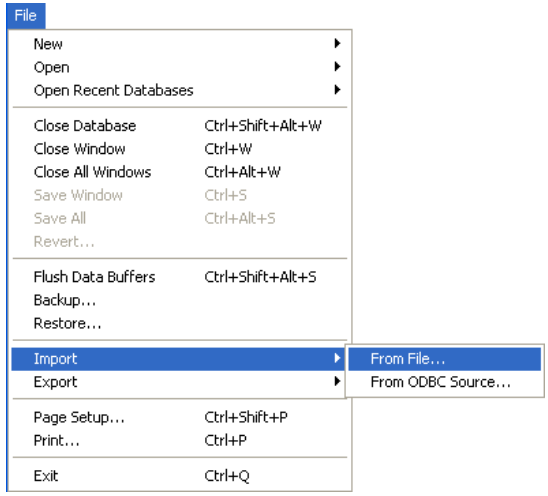
You do not need to build your entire table structure before you import data. You can create the required tables directly from the Import dialog box.

You have the option of selecting the fields for which data will be imported in the Import dialog box or specifying an input form that contains the fields for which data will be imported.

In the Design environment, you import data by the intermediary of the Import dialog box. In the Application environment, you can also display this dialog box (IMPORT DATA command) or directly import files via the commands of the "Import and Export" theme.

► To import data in the Design environment:

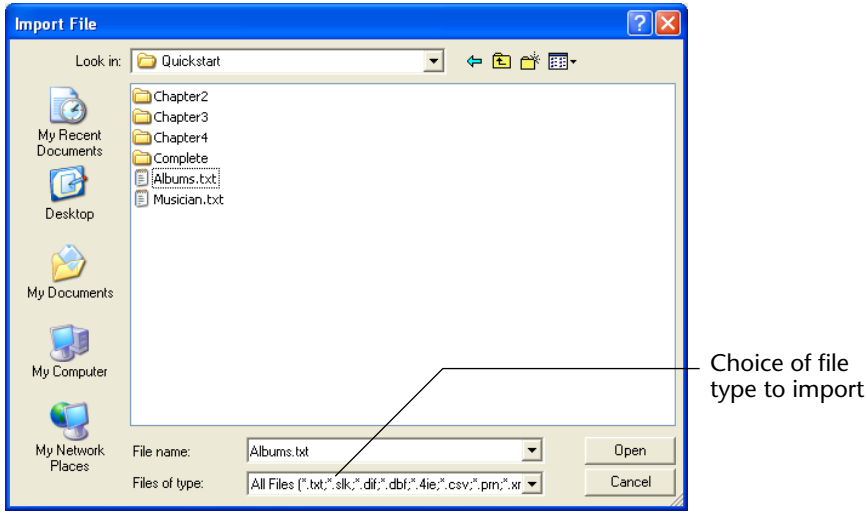
1 **Choose Import > From File... from the File menu of 4D.**



4D displays an open-file dialog box.

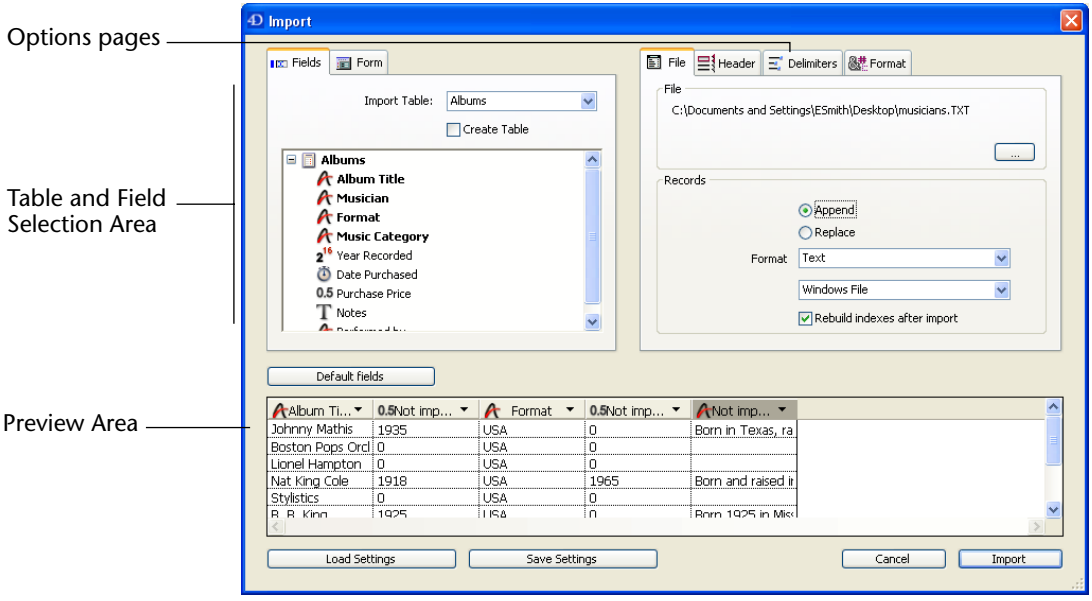
2 **Select the type of file to be imported: XML, Text, DIFF, DBF, SYLK, or 4IE (optional).**

For more information about file types, please refer to the paragraph "File Formats" on page 998.



3 **Select the file to import and click Open.**

The Import dialog box appears:



The **File** area shows the pathname of the file to be imported. If you want to import another file, click the **[...]** button to choose the file.

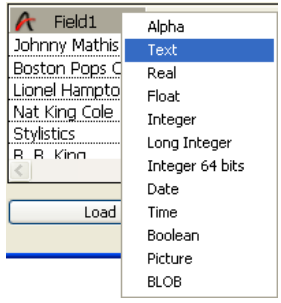
4 **Set the options of the "Records" area.**

- **Format** (type of file and *Windows File*/*Macintosh File* options). These options modify the way 4D interprets the contents of the import file. For more information about this point, please refer to File Formats, page 998 and to Field, Record, and File Delimiters, page 1000.
- **Append** or **Replace** The records imported can replace the current selection (**Replace** radio button) or be added to the existing data and form a new selection (**Append** radio button).
- **Rebuild indexes after import**: When this option is checked (by default), the indexes of the fields into which data is imported are rebuilt after the import. This mechanism can accelerate imports of large volumes of data. If the import concerns a quantity of data that is smaller than that already contained in the field, it may be useful to deselect this option before the import. In this case, the index is updated gradually and not rebuilt in its entirety.

5 **In the Table and Field Selection Area, choose the table and the fields into which you want to import the data.**

You can choose one of the following options:

- **Import data into an existing table and fields**: Select a table from the Import Table drop-down list (by default the first table is selected). The hierarchical list allows you to view and select the fields into which you want to import data.
- **Create a new table for the data import**: Check the **Create Table** check box. 4D will then create a new table into which it will import the data. 4D determines the number of fields and the type of each field according to the data being imported.
  If you want to change the name of the new table or a field, double-click on it in the Table and Field Selection area so that it becomes editable[1]. You can also change the data type for the new fields by clicking on the field's title bar in the Preview Area.
  A pop-up menu displays the different types available:



  The table is created only during the import. If you cancel the import or deselect the option, the database structure remains unmodified.

Note With the Password Access System, you can deactivate the **Create Table** option to prevent users from creating a new table in the database from the Import Data dialog box. To do so, use the Preferences dialog box to create an access group that has access to the Design environment (of course, the 4D password access system must first be activated). Users not included in this access group are not allowed to select the **Create Table** option in the Import dialog box.

- **Import data using a form**: This option is discussed in detail in the paragraph "Importing Data Using a Form" on page 1007.

1. If the Text file contains a row of field names, you can instruct 4D to use those names. For more information, please refer to the paragraph "Header Page" on page 1008.

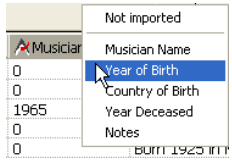
The Preview Area displays the contents of the import file as columns of data. If you import data using a form, all the fields of the form are displayed. You can resize each column.

6 **If necessary, in the Preview Area, use the pop-up menus in the header area to select the fields into which each column in the import file will be imported.**

Note This feature is not available if you checked the **Create Table** option or import data using a form.

Resizing a column affects the number of characters allocated to the column if you are using the Fixed Length Text format. Otherwise, it only affects your view of the columns.

The title bar for each column indicates the name of the field into which the column will be imported and its data type. If necessary, use the pop-up menu to select a different field:



You can also select the **Not imported** option. In this case, the column of data is not imported.

- **Default fields**
  This button allows you to "intelligently" assign columns from the import file to the fields in the table according to their type and not according to the order in which they appear. The data type of each column is estimated and it is assigned to a compatible field. The interpretation is done in the following manner:

Estimated Field Type	4D Type
All numbers	Real (Number)
True/False	Boolean
Date in one of 4D formats	Date
Time	Time
Other	Alpha

  If no compatible field is found, the column is not imported. If you want to import the column, you must assign a field to it manually in the Preview Area using the column's pop-up menu in the header area.

Note This option is not available if you use a form for the import or if you check the **Create Table** box.

- **Number of characters for the Fixed Length Text file format**: When importing a Fixed Length Text file, the Preview Area displays the number of characters for each column below the field's data type icon. The number of characters assigned to each column is based on the first row of data. You can change the distribution of characters among the columns by dragging the column dividers in the header area or entering values in the field length areas, but you can't add characters to the row.

7 **Choose any other import options using the Header, Delimiters, Filling, XML, and/or Format pages (optional).**
  These options are discussed in the paragraph "Import Options" on page 1008.

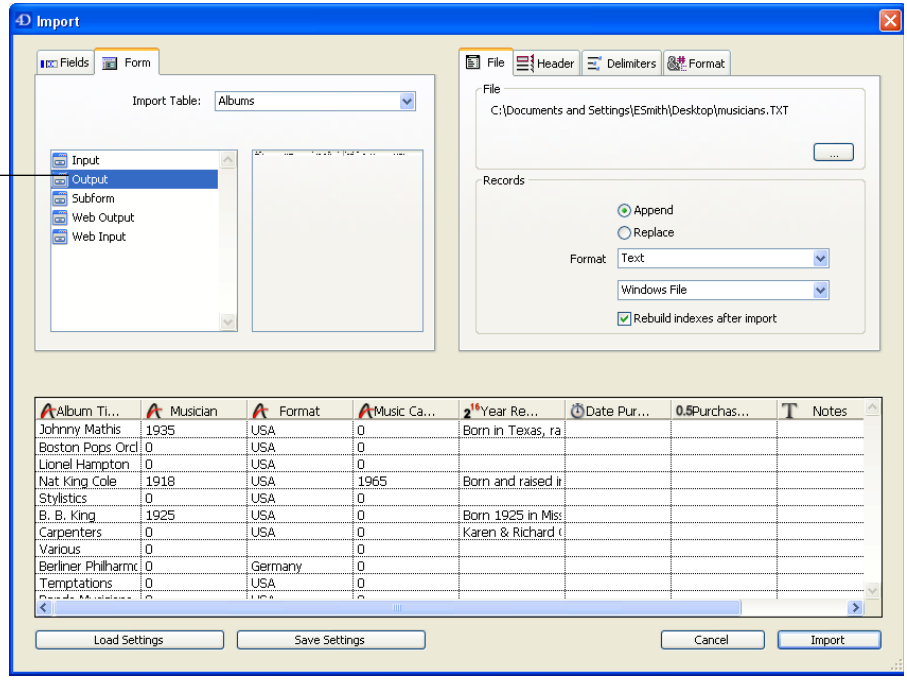
8 **If you want to save your settings, click the Save Settings button.**
  This option is discussed in the paragraph "Saving and Loading Import and Export Settings" on page 1029.

9 **Click the Import button to begin importing the text file.**
  4D displays a progress indicator as it imports the data into your database.

Importing Data Using a Form

If you want to import a text file into a form, click the **Form** tab located at the top of the Table and Field Selection Area. The advantage of using a form is that the form and object methods associated with it are executed as each record is imported. This allows you to process data while importing data (i.e., without writing a custom import routine).

The list of forms in the selected table is displayed, so that you can choose a form for the import. The data will be imported into the fields on the form using the entry order of the form.



Form selection area

You can use forms that contain enterable variables and fields, but not buttons. The form method and the methods associated with variables and fields will be executed when each record is imported. Note, however, that the effects of these methods are not shown in the Preview area.

If you use a form for the import, please keep the following considerations in mind:

- The entry order of the form determines the order in which the columns of data are imported. You should make sure that the entry order of the form matches the order of the columns in the text file.

- If there are fewer fields and variables on the form than columns in the text file, the extra columns will be ignored.
- A form used for import should not contain buttons. Also, subform objects are ignored.

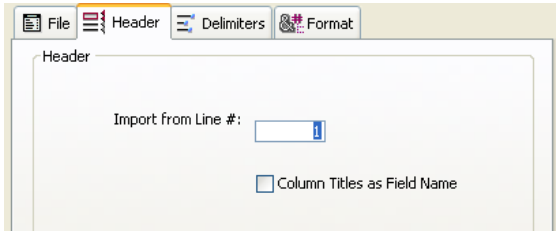
4D Server It is not possible to import data using a form that includes methods in a stored procedure on the server. In fact, since form events are not managed on the server, the associated methods will not be called.

Import Options

4D offers you various import options that are accessible through tabs in the Options area. The number of tabs as well as their contents vary and depend on the type of file used and the options selected.

Header Page

This page is available only for imports from Text and Fixed Length Text files. Two options are available:

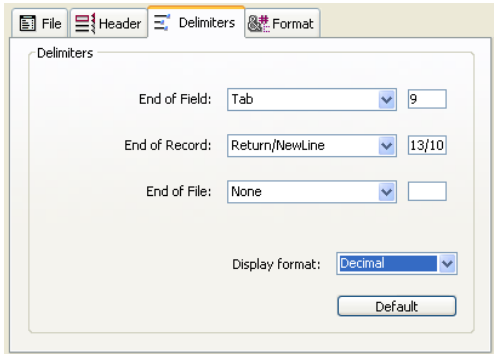


- **Import from line #**: This option allows you to specify the first line in the text file that will be imported. Use this feature to tell 4D to skip over header information — such as titles or field names — in the import file. It is particularly useful when the import file starts with unformatted lines (title, date, etc.) because the import columns are calculated according to the format of the first line.
- **Column Titles as Field Name**: This option tells 4D to use the column titles as field names if you have selected the **Create Table** option.

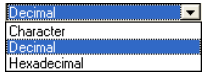
  Use these two options together, if needed. For example, if the import file starts with a title, followed by field names, followed by the first row of data, set the **Import from Line #** parameter to 2 and then check the **Column Titles as Field Names** check box.

Delimiters Page

This page allows you to specify the field, record, and file delimiters used in Text and Fixed Length Text files.



Delimiters are discussed in detail in the paragraph "Field, Record, and File Delimiters" on page 1000. To change the default delimiters, use the drop-down lists containing the values most frequently used for each delimiter. You can also enter a delimiter into the entry areas on the right.

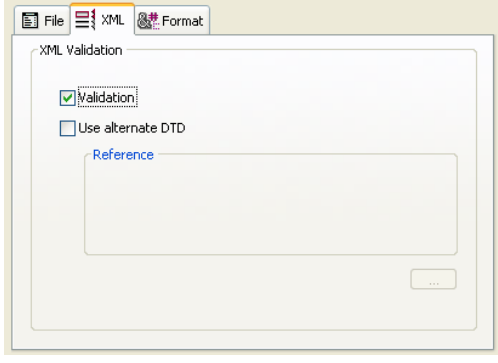


The drop-down list at the bottom of the page allows you to view the delimiters in different formats: **Character**, **Decimal** (default format), and **Hexadecimal**. Please note that, if you use unprintable characters as delimiters (such as Tab, Linefeed, and Return), the Character option will not show anything.

Note The delimiters are reset to their default values if you click the **Default** button or if the type of the document is modified on the **File** page.

XML Page

This page is available for XML files only. It allows you to configure the parsing mode of the imported XML contents.



At the time of data import, 4D interprets the contents of the XML document in order to extract the information. By default, this operation is carried out without any specific validation. The XML document is assumed to be "well-formed," meaning that its structure is correct and its interpretation is unambiguous.

You can, however, request "validation" of the document at the time of import: to do this, check the **Validation** option. In this case, 4D parses the contents of the document based on its DTD (Document Type Definition) and checks that it corresponds to this definition. Import is only carried out if the document is validated.

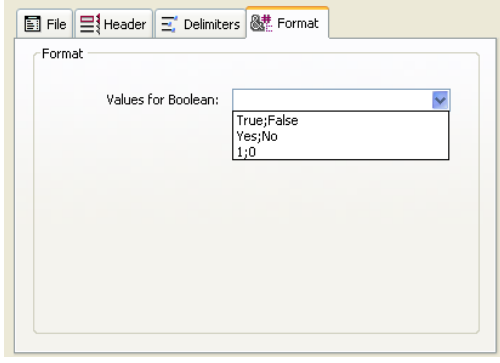
Note For more information on the DTD, refer to the paragraph "XML Page" on page 1019.

If the DTD of the imported document is not included in the document itself, but is in a separate file, or if you want to use another DTD to validate the document, check the **Use alternate DTD** option and indicate the file containing the DTD using the **[...]** button.

Once the import is completed, the dialog box is closed and the table into which the data were imported becomes the current table.

Format Page

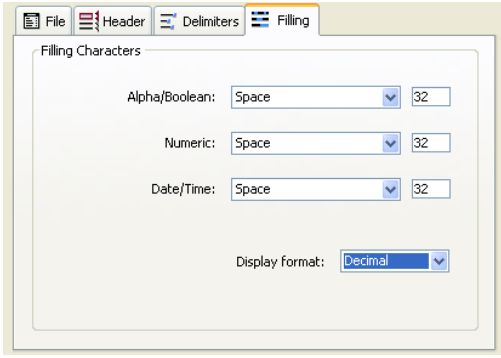
This page is available for Text, Fixed Length Text and XML files. It allows you to set the formats for imported Boolean fields.



The syntax to use is True_Value;False_Value. For example, if you import a column into a Boolean field whose values are "Black" (for true) and "White" (for false), you can enter Black;White.
The combo-box displays the most commonly used Boolean formats. You can enter any format that is not in the combo-box.

Filling Page

This page appears only for Fixed Length Text files. It allows you to define the fill characters to use for each data type. The default fill character is the space:



For more information about the Fixed Length Text file format, please refer to the paragraph “File Formats” on page 998.

Different fill characters can be used for three data types: **Alpha/Boolean** (Boolean fields are imported as alphas), **Numeric** and **Date/Time**.

Normally, the space is used to pad fixed format fields. The drop-down lists contain other choices. The entry areas to the right display the selected fill character. If you wish to use other characters, enter them directly in the entry areas.

The menu at the bottom of the page allows you to view the fill characters in different formats: **Character**, **Decimal** (default format), and **Hexadecimal**. An unprintable fill character will not appear if you use the Character format.

Exporting Data to Files

When you export data, you create a 4D, XML, SYLK, DIFF, DBF, Text, or Fixed Length Text file that can be opened by or imported by other applications. If you are exporting data for use with another application, see that application’s documentation for information about importing data. Choose a file format and delimiters compatible with the other application.

In some cases, you need to use one or more export options to structure the export file correctly for the target application. For example, some programs accept the Text file format but require that the first record consist of the field names. You can add this record using a Header option that is available in the Options Area.

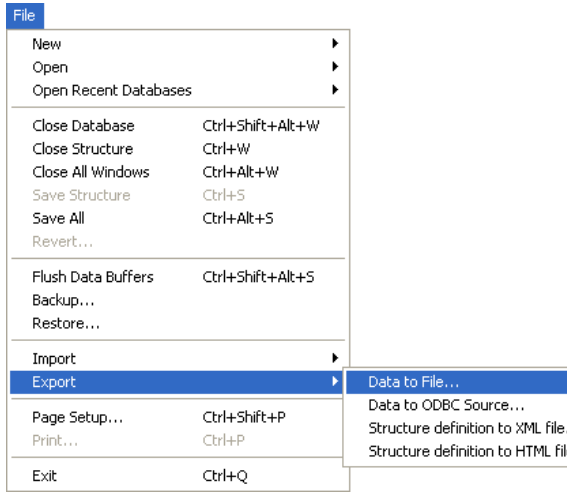
4D exports the records in the current sort order. You can choose to export all the records of a table or the current selection only.

You have the option of selecting the fields to be exported in the Export Data dialog box or specifying a form. If you use a form, the fields on the form will be exported.

In the Design environment, you export data by the intermediary of the export dialog box. In the Application environment, you can also display this dialog box (EXPORT DATA command) or export files directly via the commands of the "Import and Export" theme.

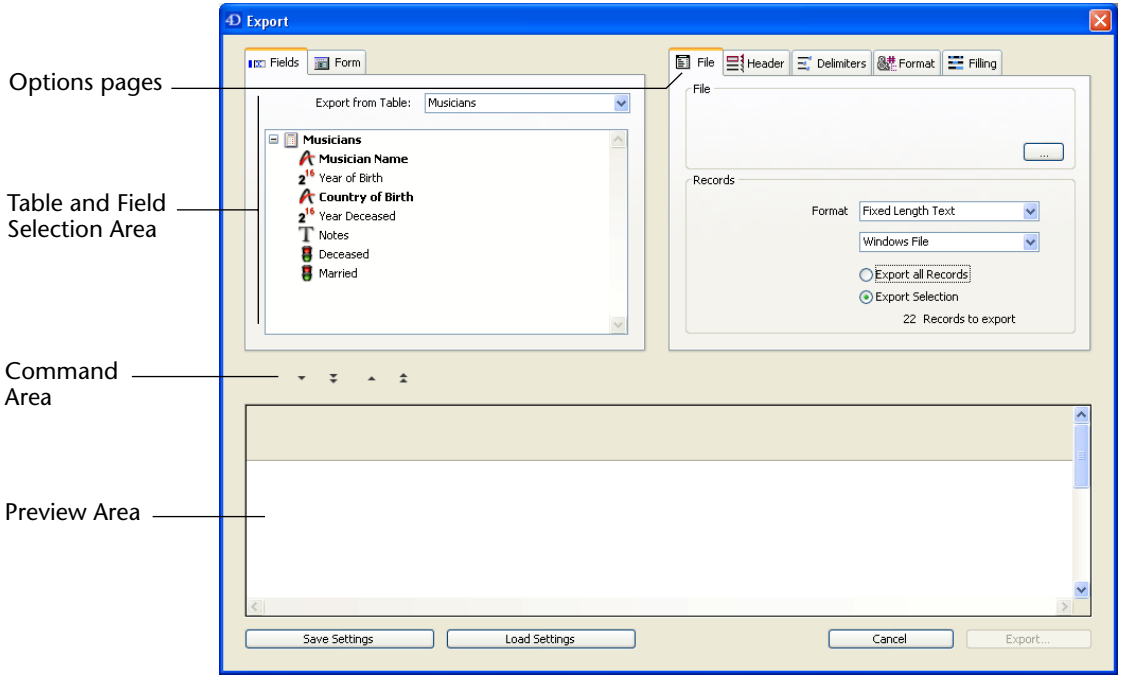
► To export data in the Design environment:

1 Choose Export > Data to File... from the File menu of 4D.



The Export Data dialog box appears.

It consists of the following items:



2 **In the "Records" area, select the format of the export file — Text, Fixed Length Text, DIFF, SYLK, DBF, 4D Application or XML and its type — Macintosh File or Windows File.**

For more information, please refer to the paragraph "File Formats" on page 998.

3 **Click the [...] button, enter a name and a location for the export file and click Save (optional).**

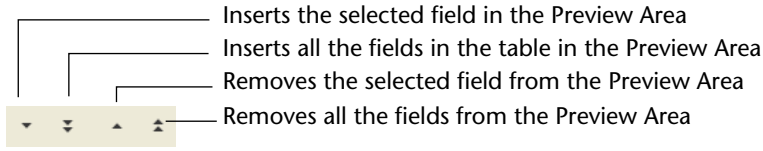
The Export Data dialog box reappears, with the export file's pathname shown in the File area. *This step does not start the export.*

Note If you skip these steps, you can also specify the name and location of the export file when you click the **Export** button.

4 **In the Table and Field Selection Area, select the table and the fields that you want to export.**

You can choose either of the following options:

- **Select the fields to export in the Export Data dialog box**
  If you use this option, choose a table from the Export From Table drop-down list to populate the Fields list with the eligible fields for the export. You then choose fields using the two blue buttons just above the Preview area. You can remove fields from the Preview area using the red buttons:



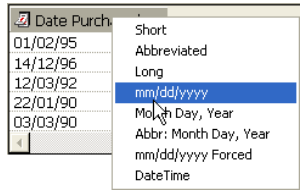
- **Export the data using a form**
  This option exports the fields on the form. This option is discussed in detail in the paragraph “Importing Data Using a Form” on page 1007.

5 In the Preview Area, modify the fields from which the data will be exported (optional).

Note This feature is not available if you export data using a form.

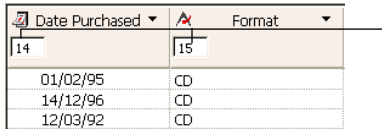
Several modifications can be made using the Preview Area:

- **Choosing other fields**: The Preview area displays the contents of the export file as columns of data. The Header area for each column is a pop-up menu that you can use to change the field assigned to that column.
- **Specifying the export format**: The icon to the left of the field name indicates its data type. You can right-click to display a contextual menu of formats that are appropriate for the column’s data type. If desired, choose a format from the contextual menu.



If you do not assign a format, the Default Format is used. A format can also be selected on the Format options page. For more information, see the paragraph “Format Page” on page 1010.

- **Changing the lengths of Fixed Length Text fields**: If you choose the Fixed Length text format, the Preview Area indicates the number of characters for each column. You can resize each column by dragging the column divider in the title bar with the mouse. If you are using the Fixed Length Text format, this changes the number of characters allocated to the column in the export file. You can set the exact number of characters by entering a value in the entry area below the field's data type icon. The default number of characters is as follows: Text fields have a length of 80 characters, Alpha fields have the maximum length assigned in the Design environment, and numeric fields have a length of 10 characters.

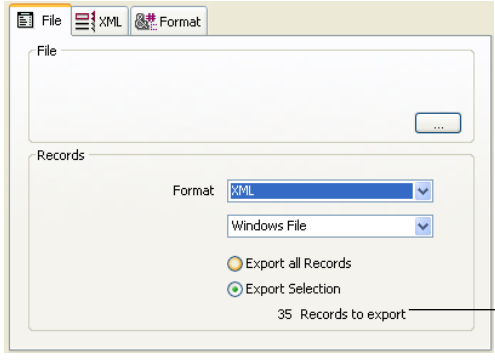


The fill characters are added at the end of alphanumeric fields and at the beginning of numeric fields.

- **Removing fields from the export file**: You can remove a column from the export file and the Preview Area. To do so, click the column's header and click the **Delete** button in the button panel  . To delete all the fields, click the **Delete All** button in the button panel  .

6 **In the Records area, indicate if you want to export all the records of the selected table (the Export all Records option), or only the current selection (the Export Selection option).**

In both cases, the number of records to be exported is displayed in this area.



7 **Specify any other export options using the Header, Delimiters, Format, XML, and Filling pages (optional).**

These options are discussed in the paragraph “Export Options” on page 1016.

8 **If you want to save your settings, click the Save Settings button.**

This feature is discussed in the paragraph “Saving and Loading Import and Export Settings” on page 1029.

9 **Click Export to begin exporting to the disk file.**

If you have not specified the pathname for the export file, a standard save file dialog appears (see step 3).
Otherwise, the export is carried out directly.

Exporting Data Using a Form

If you want to use a form for the export, click on the **Form** tab, located at the top of the Export dialog box. You can then choose a form from the list of forms for the selected table.

The main advantage of using a form is that the form method and the object methods associated with fields and variables are executed when each record is exported. This allows you to process the data during the export. Fields and variables will be exported in the order specified by the entry order of the form. Note, however, that the effects of these methods are not shown in the Preview Area.

Do not place buttons on the form. Subform objects are ignored.

4D Server It is not possible to export data using a form that includes methods in a stored procedure on the server. In fact, since form events are not managed by the server, the associated methods will not be called.

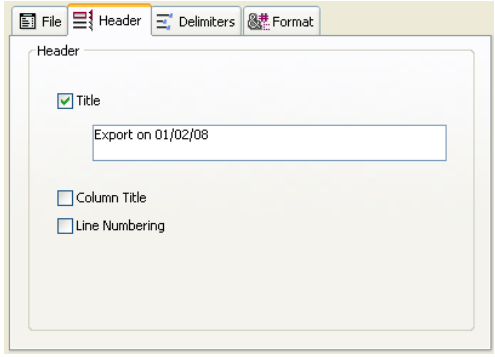
Export Options

4D offers you various export options that are accessible using tabs in the Options area. The number of tabs as well as their contents depends on the type of export file and the selected options.

This section describes all the Export options pages; some may only be available for certain export file types.

Header Page

This page is only available for Text and Fixed Length Text exports.



It allows you to add a title, row of field names, and/or a record number to the export file:

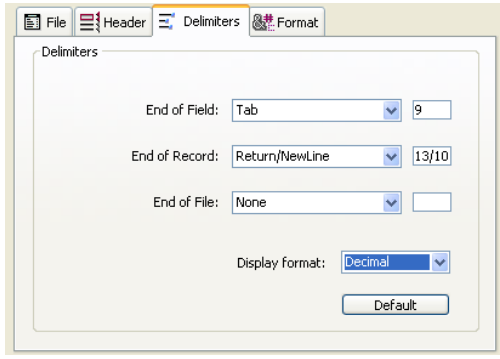
- **Title**: This option allows you to enter a title for the export document. Enter the title in the area below the check box.
- **Column Titles**: This option exports the field names as the first 'record' in the export file.
- **Line numbering**: This option numbers each line, which is each exported record. The numbering starts at 1 and increments by 1.

If **Title** is selected, it is the first row in the export file even if Column Titles is also selected. If **Column Titles** is selected, the field names follow the title (if also present) and precede the first row of data; for Text files, the field names are separated by the field delimiter.

If **Line Numbering** is selected, the line number precedes the first field in each row, i.e., the first field as specified in the Preview Area becomes the second field, and so on.

Delimiters Page

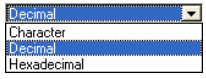
This page allows you to specify the delimiters used in Text and Fixed Length Text files, so that the application importing the data can interpret it correctly. It is not relevant for other types of export files.



Delimiters are discussed in detail in the paragraph “Field, Record, and File Delimiters” on page 1000.

To modify the delimiters, you can use the drop-down menus to the left of each area. These menus contain the values most frequently used for each type of delimiter.

You can also enter a delimiter directly in the area on the right.

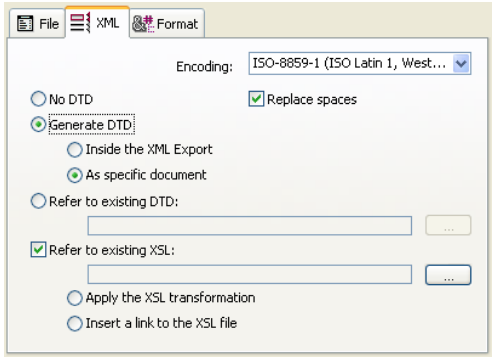


The menu at the top of the page allows you to modify or view the delimiters in different formats: **Character**, **Decimal** (default format), and **Hexadecimal**. Please note that if you use an unprintable character as a delimiter (such as a tab, linefeed, or return), the Character option will not show anything. The Decimal and Hexadecimal formats can display any character’s ASCII code.

Note The delimiters are reinitialized to their default value if you click the **Default** button or if the type of the document is modified on the **File** page.

XML Page

This page of parameters is only available for XML exports. It enables the configuration of the exported XML file contents:



- **Encoding**
  This pop-up menu is used to select the encoding (i.e. the set of characters) to be used for the XML document. The choice of encoding depends on the contents of the data to be exported and the application that will be using them. By default, ISO-8859-1 (ISO Latin 1, West European) coding is selected.

- **Replacing spaces**
  The **Replace spaces** option causes "space" characters to be replaced by underlines ("_") in value name fields of the XML file generated. This option is checked by default since spaces are not allowed in XML value name fields.
  However, it is possible to uncheck the option when necessary for specific purposes. In this case, of course, the generated file will not be in conformity with the general syntax rules defined by the W3C for XML.

- **DTD options**
  When exporting in XML format, 4D allows you the choice of whether or not to generate a DTD (Document Type Declaration). A DTD records the set of specific rules and properties that the XML must follow. These rules define, more particularly, the name and contents of each tag, as well as its context.

  This formalization of elements enables you to check an XML document to make sure it is "valid" and is particularly useful in the case of recurrent tags in an XML document. Note that a DTD is not mandatory.

To define the handling of the DTD, you must select one of the three following options:

- **No DTD** (default option): When this option is selected, a DTD is not generated during export.
- **Generate DTD**: generates a DTD during export. When this option is checked, the following subset of radio buttons enables you to specify the location of the DTD to be generated:
  - ***Inside the XML Export***: A DTD will be included in the XML file itself (internal DTD). The generated XML file is thus independent.
  - ***As specific document***: A DTD will be generated in a separate file (external DTD). An external DTD can be shared between several users and thus enables the harmonization of XML document structures generated from different sources.

Note 4D allows a table and a field to have the same name. However, the XML language prohibits the use of different elements having the same name. Consequently, when the "Generate DTD" option is used, the exported 4D data must not include a table and field having the same name; otherwise, the XML file generated is invalid and cannot be opened by an XML parser.

- **Refer to existing DTD**: Using the associated **[...]** button, this option allows you to designate an existing external DTD file. 4D will include a reference to this DTD in the exported file.

- **Refer to existing XSL**
  XSL (eXtensible Stylesheet Language) enables visual representation of the elements defined in an XML document. This language is used to schematically define the style sheets that can be used to process and display the contents of an XSL document. Using the **[...]** button, the **Refer to existing XSL** option allows you to associate an XSL file for the exported data.

Note For more information about the support of XSL, refer to the *4D Language Reference* manual.

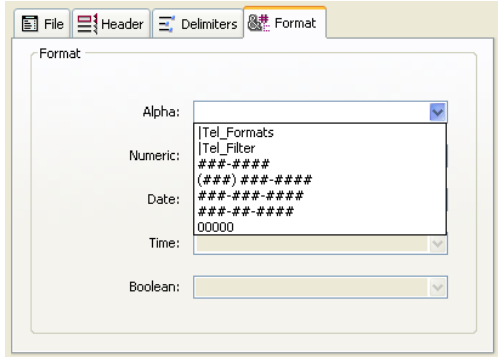
When an XSL file has been selected, you must set up how it will be used with exported data using two options:

- **Apply the XSL transformation**: When this option is checked, 4D performs the transformation of XSL data directly on the set XSL document at the moment of export.

- **Insert a link to the XSL file**: When this option is checked, 4D only inserts the reference to the designated XSL in the exported document (the transformation is not carried out).
  The exported XML file will therefore contain references to this XSL file.

Format Page

This page allows you to set the export formats. It is available for all the files formats, except for the 4D Application format. This option is available for Text and Fixed Length Text exports. By default, 4D's standard formats are used.



Note You can also set a column's export format using the contextual menu in the column's header area (see step 5 in the paragraph "Exporting Data to Files" on page 1011).

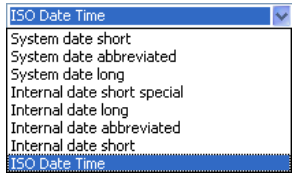
► To assign an export format:

1 **Select a column in the Preview Area.**
The combo-box corresponding to its data type becomes enabled.

2 **Choose a format from the combo box or (for Alpha, Numeric, or Boolean data types) enter a custom format.**

Note For more information about 4D display formats, please refer to the paragraph "Display Formats" on page 521.

- **About ISO Date Time Format**



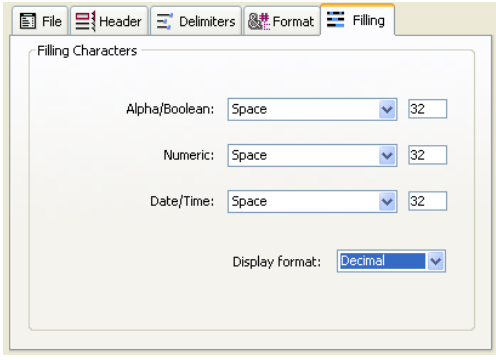
The **ISO Date Time** display format is available for exporting date or time data. This format corresponds to the XML date and time representation standard (ISO8601 format). For example, in this format the date/time May 31, 2008 at 1:20 p.m. is noted 2008-05-31T13:20:00.

4D does not allow both a date and time to be stored in a single field. However, you can export data in this format so that the dates or times are recorded in conformity with the XML standard. If you export dates, the exported values will be of the style: 2008-05-31T00:00:00; whereas if you export times, they will be of the style: 0000-00-00T13:20:55.

Note You can import XML data recorded in this format. They can be stored either in a date or time field, depending on the information that you want to keep.

Filling Page

This page is only relevant for Fixed Length Text exports. It allows you to define the fill character to use for each data type. You can specify a different fill character for Alpha/Boolean (Boolean fields are exported as Alpha fields), Numeric (Integer, Longint, and Real), and Date/Time values.



For a description of the Fixed Length Text file format, please refer to the paragraph “File Formats” on page 998.

Most often, the Space character is used. The drop-down lists contain other possible values. The entry areas to the right display the characters selected. If you wish to use other characters, enter them directly in these areas.

The drop-down list at the bottom of the page allows you to view the fill characters in different formats: **Character**, **Decimal** (default format), and **Hexadecimal**. Please note that an unprintable character will be invisible if you select the **Character** format. The Decimal and Hexadecimal formats can display any character's ASCII code.

Importing and Exporting Data via an ODBC Data Source

4D lets you import and export data from or to an ODBC data source.

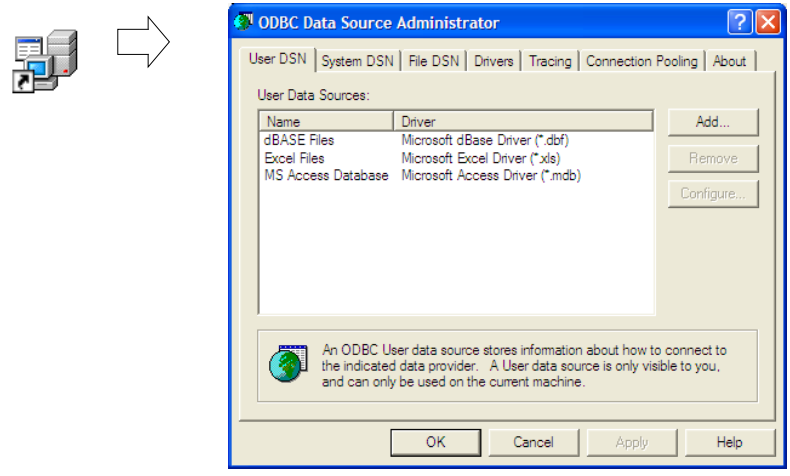
What is an ODBC Source?

The ODBC protocol (Open DataBase Connectivity) is an interface allowing different applications to communicate between each other using SQL language. An ODBC data source is a permanently open "window" to a document belonging to an application, even when this application is not open. Many applications (databases, spreadsheets, data managers, etc.) allow publishing data as an ODBC source using a specific ODBC driver.

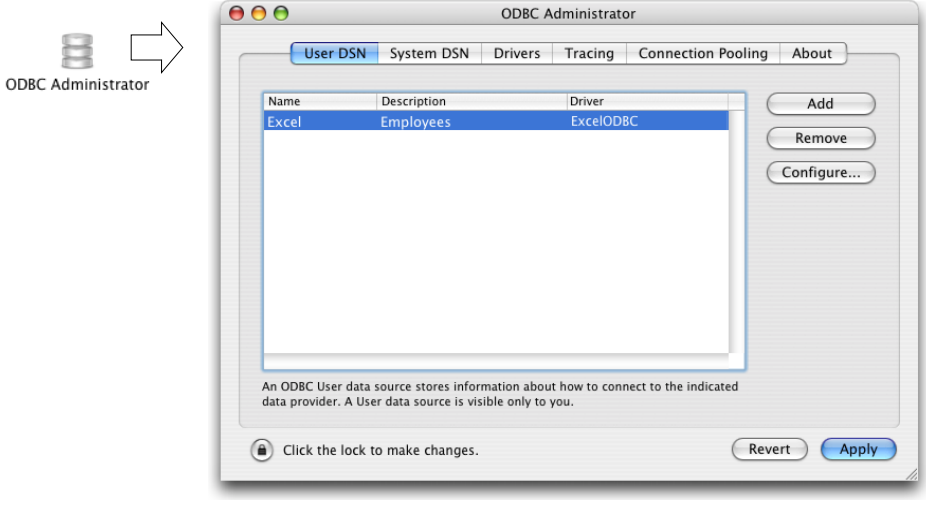
Any application supporting the ODBC protocol can read or write data in an ODBC source. The 4D import function allows getting data from an ODBC source while the export function allows inserting data into the ODBC source.

Defining an ODBC Source

- Under Windows, the definition of an ODBC source is carried out using the **Administrative Tools** then **Data Sources (ODBC)** configuration panel:



- Under Mac OS X, the definition of an ODBC source is carried out using the **ODBC Administrator** program located in the **Utilities** folder:



Note For more information on these dialog boxes, please refer to the operating system documentation.

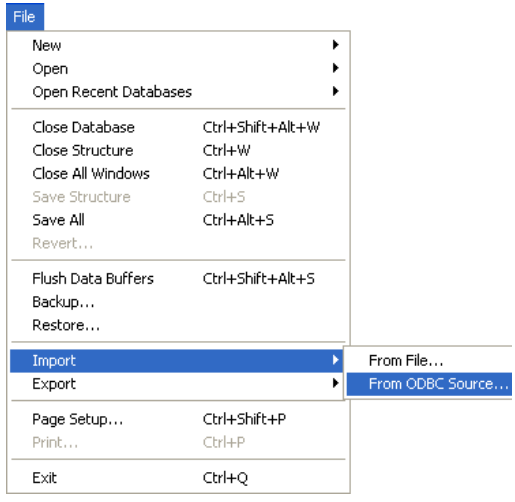
Importing Data from an ODBC Source

The concept of importing data from an ODBC source is the same as when importing other types of data, with the exception that an intermediary dialog box allowing you to select an ODBC source is used.

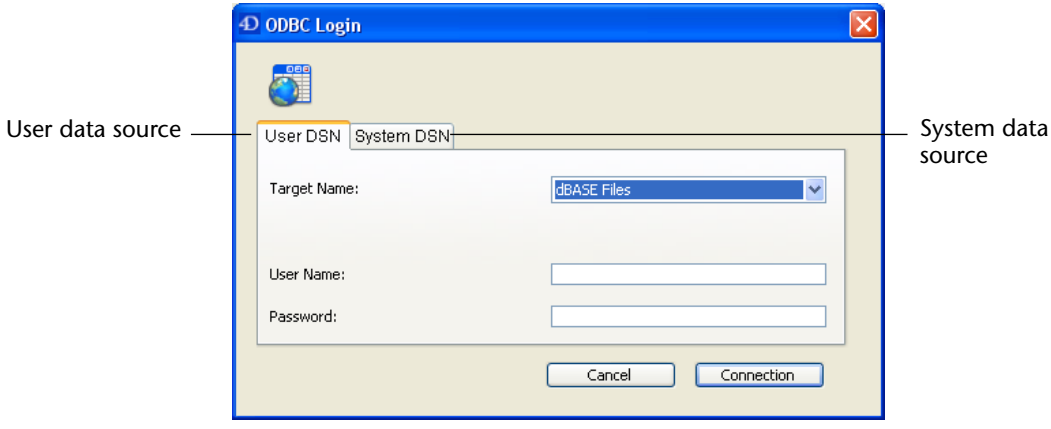
In the Application environment, importing can be carried out using the ODBC IMPORT command. In the Design environment, it is done using the **File** menu of 4D.

► To import data from an ODBC source:

1 Choose the Import > From ODBC Source... command in the 4D File menu:



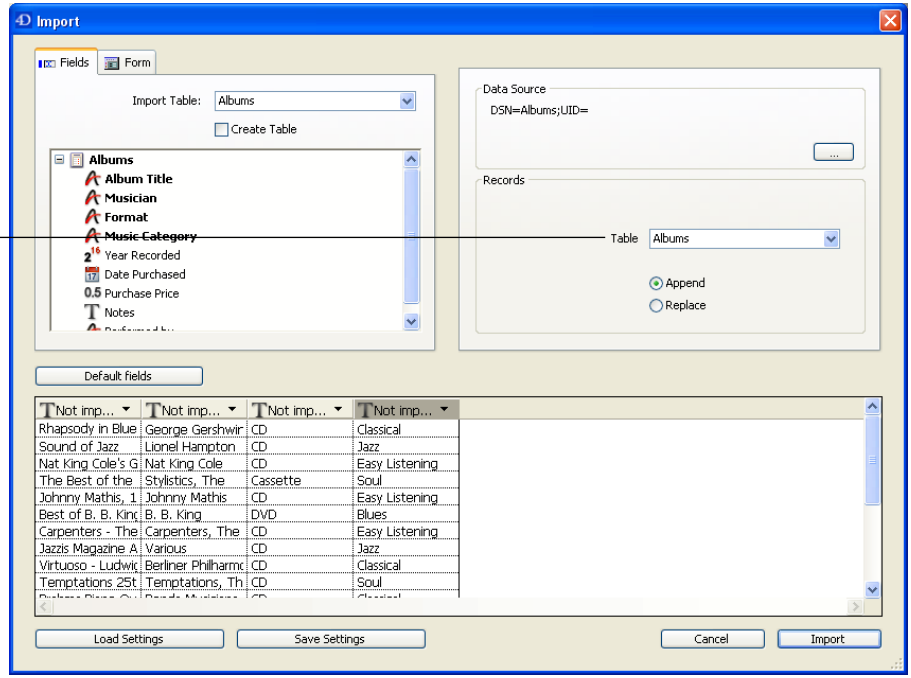
The ODBC login dialog box appears:



2 **Select the data source to use as well as (if necessary) the access IDs then click Connection.**

The standard 4D import data dialog box appears.

3 **On the right side of the dialog box, set the SQL table from which you wish to import data:**



Selection of SQL table

4 **Select the table and 4D fields into which the data must be imported or check the "Create a table" option (this option is not available for compiled databases).**

Like all other kinds of imports, the preview area displays matches between imported data and fields. You can also click the **Form** tab and import data using a form. For more information about these functions, refer to steps 5 and 6 in the paragraph "Importing Data from Files" on page 1001 as well as the paragraph "Importing Data Using a Form" on page 1007.

5 **Choose import options on the right side of the dialog box:**

- **Data Source**: This area describes the data source to which the database is connected. You can display the select data source dialog box again by clicking **[...]**.
- **Records**: You can add or replace the records of the import table with data coming from the source.

6 Click Import to import the data.

Once the data is imported, the connection between 4D and the data source is automatically closed.

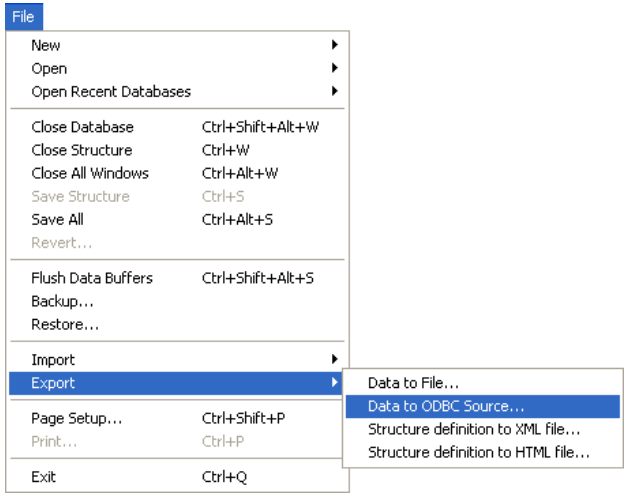
Exporting Data to an ODBC Source

The concept of exporting data to an ODBC source is the same as when exporting other types of data, with the exception that an intermediary dialog box allowing you to select an ODBC source is used. Also, exported data is not stored in a disk file but directly in the published ODBC data source.

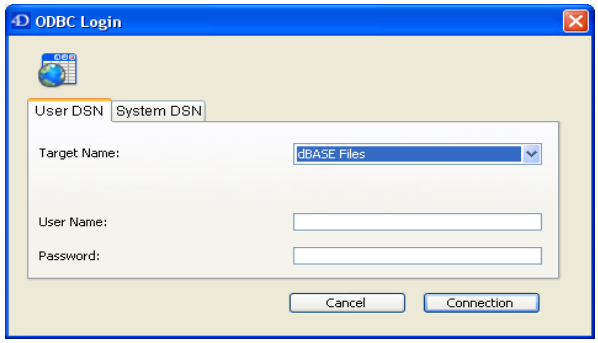
In the Application environment, exporting can be carried out using the ODBC EXPORT command. In the Design environment, it is done using the **File** menu of 4D.

► To export data toward an ODBC source:

1 Choose the Export > Data to ODBC Source... command in the 4D File menu:



The ODBC login dialog box appears:



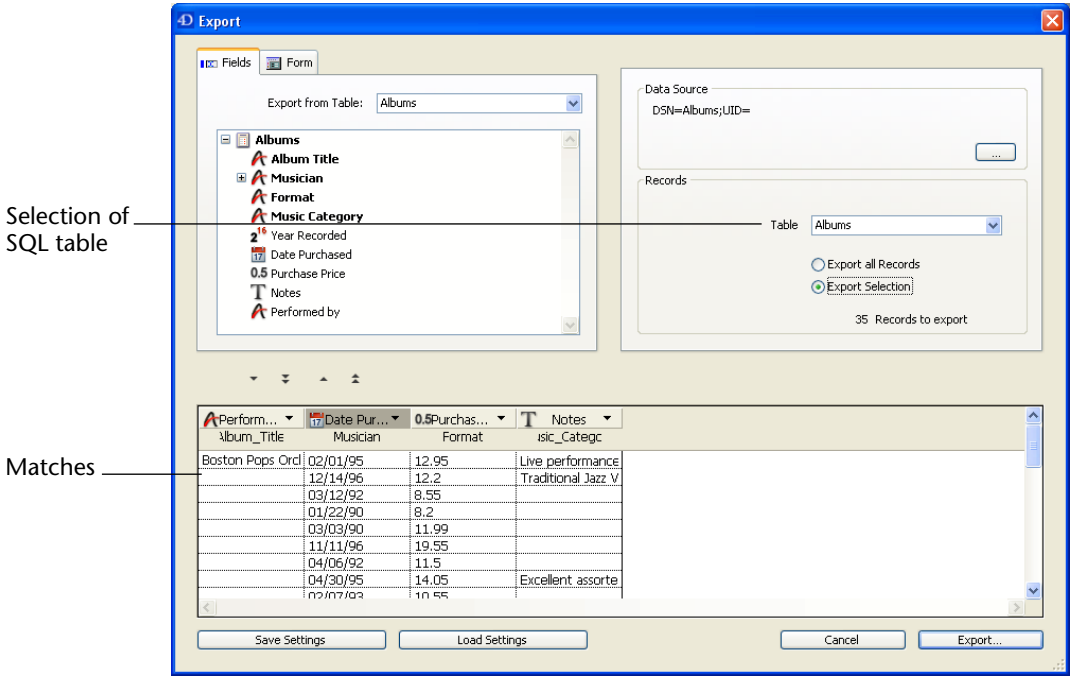
2 **Select the data source to use as well as (if necessary) the access IDs then click Connection.**

The standard 4D export data dialog box appears.

3 **On the right side of the dialog box, set the SQL table into which you wish to export data.**

4 **Select the table and 4D fields from which the data must be exported.**

Like all other kinds of exports, the preview area displays matches between exported data and fields. You can modify these matches.



You can also click the **Form** tab and export data using a form. For more information about these functions, refer to steps 4 and 5 in the paragraph “Exporting Data to Files” on page 1011 as well as the paragraph “Exporting Data Using a Form” on page 1016.

5 **Choose export options on the right side of the dialog box:**

- **Data Source**: This area describes the data source to which the database is connected. You can display the select data source dialog box again by clicking **[...]**.
- **Records**: You can only export the current selection or all records of the table.

6 **Click Export... to export data toward the ODBC source.**

Once the data is exported, the connection between 4D and the data source is automatically closed.

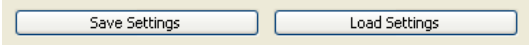
Saving and Loading Import and Export Settings

The import and export dialog boxes allow you to save and load your settings to and from disk. A settings file stores all the settings specified in the Import or Export editor:

- Name and access path of file or ODBC source,
- Tables and fields selected and the name of the form, the export uses a form,
- Import and export options (file type, delimiters, etc.).

You can carry out this operation using the import-export dialog boxes of files and ODBC sources. However, projects saved in the ODBC import-export dialog box are not compatible with the standard import-export dialog box and vice versa.

To save or load the import or export settings, click on the **Save Settings** or **Load Settings** buttons at the bottom of the dialog box:



Import and export settings files have the extension “.4SI” on Windows (they are of “4DSI” type on Mac OS).This feature allows you to automate the import or export process. Using a settings file is particularly useful in conjunction with the IMPORT DATA and EXPORT DATA commands. For more information, please refer to the *4D Language Referen*ce manual.

Note An import/export settings file does not store filters because they are also related to the serial communication. If necessary, you will need to load an import or export filter before the operation.

20 Labels

4D's Label editor provides a convenient way to print a wide variety of labels.

With the Label editor, you can do the following:

- Design labels for mailings, file folders and file cards, and for many other needs
- Specify the font, font size, and style to be used for the labels
- Specify the number of labels across and down on each page
- Specify the label page margins
- Load and save label designs
- Print labels.

Labels can also be created using the Form editor. Use the Form editor to design specialized labels that include variables or take advantage of the drawing tools available in the Form editor. For more information about using the Form editor to create labels, refer to the paragraph "Printing Labels" on page 680.

The Label Editor

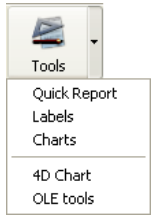
You use the Label editor to create, format, and print labels. The Label editor contains settings for designing labels and positioning the labels on label paper. For example, when producing mailing labels, you might want a label design that includes the person's first and last name on the first line, the street address on the second line, and so on.

As part of the design, the Label editor enables you to specify the number of labels on the page and the margins of the label paper so that the label text is centered within the labels.

The Label editor consists of two pages, the Label page and the Layout page, each identified by a tab at the top of the dialog box. You use the Label page to specify the contents of the label and the Layout page to define the size and position of the labels on the page. When you create a satisfactory label design, you can save it to disk so that you can reuse it.

► To open the Label editor:

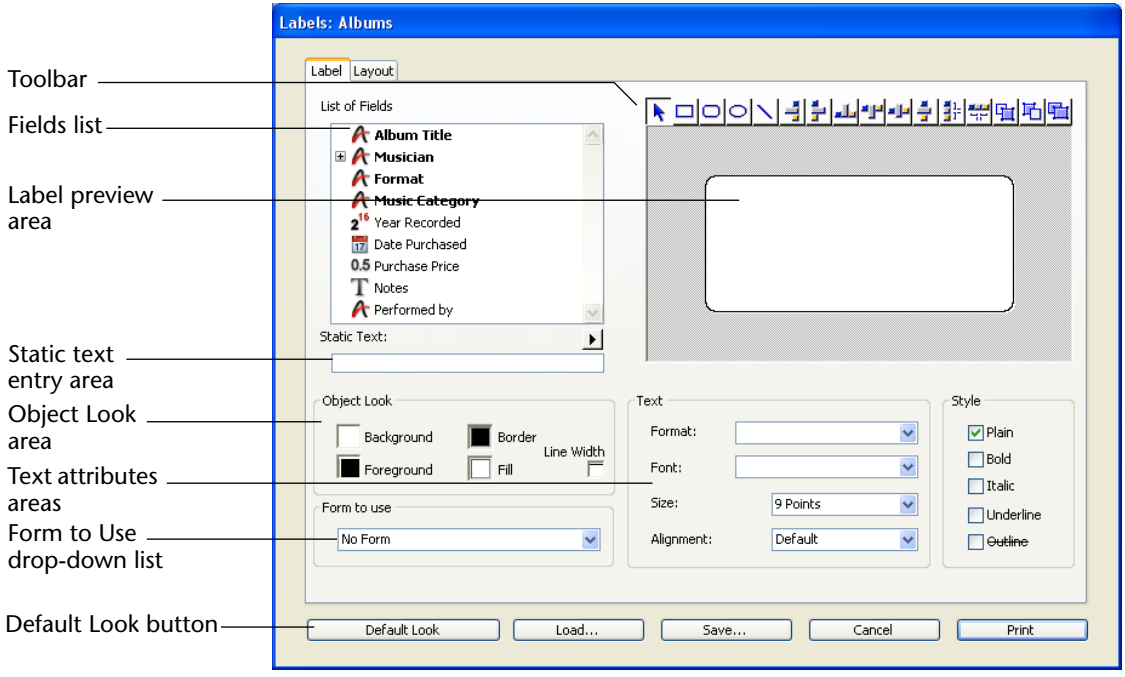
1 **Choose Labels... from the Tools menu or select Labels... from the menu associated with the "Tools" button in the 4D tool bar.**



OR

In the Application environment, use the PRINT LABEL command.

4D displays the Label page of the Label editor.



Label Page

The Label page contains settings for designing and formatting labels.

It contains the following elements:

- **Fields list**: This area displays the names of the fields in the current table in a hierarchical list. If this table is related to other tables, the foreign key fields have a plus sign (on Windows) or an arrow (on Macintosh). You can display fields from the related table by expanding the related fields. The fields in the related table are indented.

Note Only tables and fields which are visible appear in the Label editor. For information about making tables and fields invisible, refer to the chapter "Database Structure" on page 239.

- **Label preview area**: You use this area to design your label.
- **Toolbar**: The Label editor toolbar contains tools for drawing, selecting, aligning, distributing, layering, and duplicating objects. For complete information on the toolbar, see the chapter "The Label Editor Toolbar" on page 1034.
- **Object Look area**: These controls let you specify foreground and background colors, fill patterns, and borders for individual objects on the label.
- **Default Look button**: This button applies the default set of Object Look attributes to the selected object.
- **Static Text entry area**: This area allows you to add static text objects to the label.
- **Text attributes areas:** These controls allow you to specify the font, font size, display format, and style of the text.
- **Form to Use drop-down list**: This drop-down list lets you bypass the Label editor and use a form to print the labels. If you are using the Label editor to create the label, choose No Form (default) from this list. If you want to use a form, choose it from this list. 4D will then ignore any other label specifications in the Label editor and print the labels according to the design of the specified form. As with any print job, it executes any form or object methods associated with the form. See the paragraph "Printing Labels" on page 680 for more information about designing forms for labels.

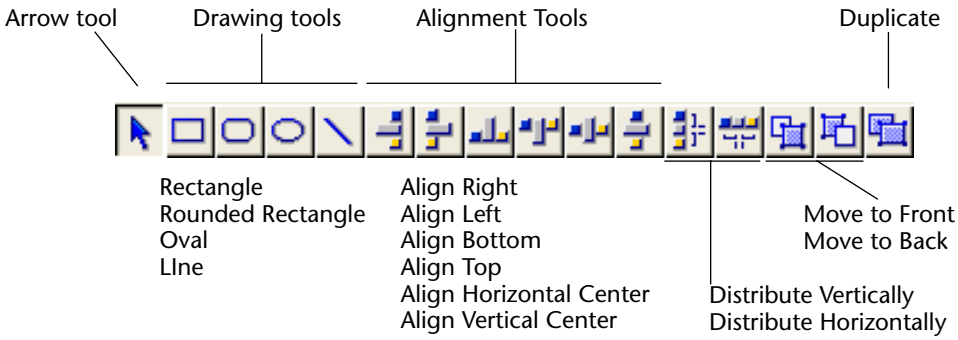
Layout Page

The Layout page contains controls for printing labels based on the requirements of the printer you selected in the Print Manager (Chooser on Macintosh), so that it can format the page accurately.

For complete information on the Layout page, see the chapter "Specifying the Label Layout" on page 1045.

The Label Editor Toolbar

The Label editor toolbar contains the following tools.



Creating a Label Design

You create a label by dragging field names to the Label Preview area, adding static text, drawing graphic objects using the graphics tools in the toolbar, and pasting graphics from the Clipboard. You can edit the label by distributing, moving, resizing, layering, and aligning objects. With the Object Look and Text attribute areas, you can control the font attributes, foreground and background colors, fill patterns, and borders of individual objects.

The Label Preview area shows the approximate size and shape of the selected label. Before adding elements to the label, you may want to specify your label paper and label dimensions using the Layout page. For information, see the chapter "Specifying the Label Layout" on page 1045.

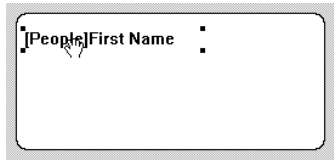
Note You cannot add a formula (a variable) to a label with the Label editor. If you need to use a variable on a label, use the Form editor.

► To create a label:

1 Drag the first field you want to display in the label from the Fields list to the Label Preview area.

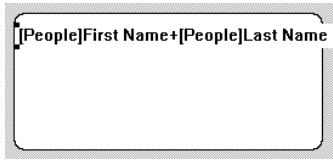
If the field is in a related table, expand the related field to display the fields in the related table.

Your field is added to the Label Preview area. Selection handles indicate that it is selected.

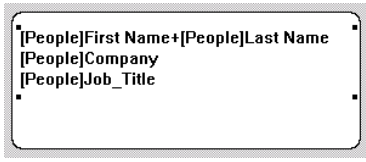


2 If you want to concatenate a field to this field, drag the new field from the Field list to the existing field.
OR
If you want to concatenate two fields using the Carriage return as the separator, drag the new field by holding down the Shift key.
OR
Otherwise, continue dragging fields to the Label Preview area.

- When you concatenate two fields by simply dragging and dropping them, the separator used is the one defined in 4D's resources. By default, the Space character is used. A "+" between field names in a field object indicates that the fields are concatenated on a single line. When 4D prints the label, it will insert a space between the fields on the same line. The following illustration shows the concatenation of the First Name and Last Name fields.



- If you concatenate two fields by dragging and dropping the second field while holding down the Shift key, the Carriage return is used as the separator. When the labels are being printed, if the second field is empty, its position is deleted and is replaced by the next field. If a field contains some text with Carriage returns, they are taken into account.



  For example, this feature allows you to create address labels by using many concatenated fields ([Clients]Address1, [Clients]Address2, etc.) without generating an empty line when a field is blank.
  You can automatically resize a group of concatenated fields so that it corresponds to the number of lines that it contains. To do so, **Ctrl+click** (Windows) or **Command+click** (Mac OS) on the lower right handle of the group.

As you add fields, you can reposition them by dragging or using the alignment tools in the toolbar.

3 **To add a text element to the label, enter the text in the Static Text area and click the arrow** ▶**.**

The static text object is added to the Label Preview area. The following illustration shows a static text element being added to the label:

After you add the element to the label, you can reposition it by dragging and aligning it with other objects.

4 **(Optional) Using a drawing tool, draw any graphic objects that you want to add to the label.**

For example, you could add different backgrounds to the "TO:" and "FROM:" sections of the label.

5 **(Optional) Paste a graphic from the Clipboard into the Label Preview area.**

For information on adding a graphic to the label, see the chapter "Adding a Graphic to the Label" on page 1044.

After you add all required elements to the Label Preview area, you can use any of the editing tools to modify the label design. For complete information, see the chapter "Working with Label Editor Objects" on page 1037.

How the Label Editor Handles Blank Fields

Some records in your database may not contain entries for every field. When printing labels, the Label editor handles blank fields intelligently. Instead of leaving gaps where the data should be, the Label editor concatenates the data.

If a field in a line contains no data for a particular record, 4D concatenates the remaining fields in the line without leaving a blank space for the missing data.

If an entire line contains no data for a particular record, 4D vertically concatenates the remaining lines in the label without leaving a blank line.

4D automatically centers the text of the label inside the label area.

Clearing Fields

If you make a mistake, you can remove one or more fields from the label.

▶ To remove an object:

1 **Select the object and press Backspace (Delete on Macintosh).**

The selected object is removed. If the object is a row of concatenated fields, pressing Backspace removes only the last field in the row. Continue pressing Backspace to remove more fields from the row.

Working with Label Editor Objects

This section discusses the techniques for editing objects placed on the label. It includes:

- Creating graphic objects,
- Aligning objects,
- Distributing objects,
- Layering objects,

- Duplicating objects,
- Moving objects,
- Resizing objects,
- Adding a border to an object,
- Adding foreground or background fill colors, patterns, and borders,
- Pasting a graphic into the label,
- Deleting objects.

Creating Graphic Objects

You can create a graphic object by drawing.

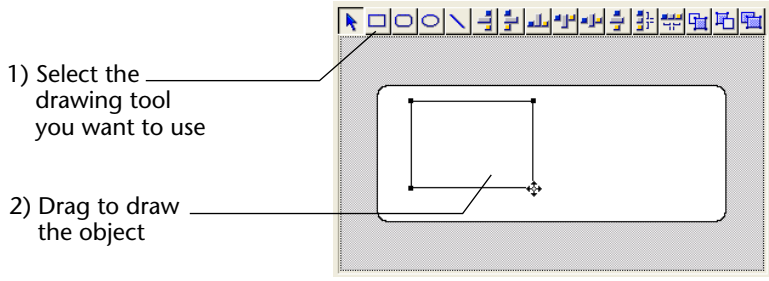
► To create an object:

1 **Select the type of object you want to draw by clicking its tool in the toolbar.**

The pointer becomes a crosshair when it is over the Label Preview area.

2 **Drag to create an area for the object.**

For two-dimensional objects (ovals, rectangles, and rounded rectangles), drag diagonally.



Note Hold down the **Shift** key as you draw to constrain the object to a regular shape. Lines are constrained to horizontal or vertical, rectangles are constrained to squares, and ovals are constrained to circles.

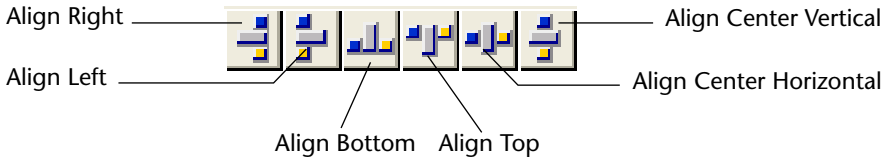
3 **When you have finished drawing the object, release the mouse button.**

4D creates the object and makes it the currently selected object. The Arrow tool is automatically selected and the pointer becomes an arrow.

Aligning Objects

The alignment tools let you align objects to each other. When you align one object to another, you can align it to the top, bottom, side, or horizontal or vertical center of the other object.

The following illustration shows the Label editor's alignment tools.



The arrangement of lines represents the function of each tool. For example, the Align Right icon shows the lines aligned vertically on the right side of the box. The Align Center Vertical icon shows the lines aligned vertically in the middle.

► To align a set of objects:

1 **Select the objects that you want to align.**
Shift-click to select several objects.

2 **Click the alignment tool that corresponds to the alignment you want.**
4D aligns the selected objects according to the alignment you selected.

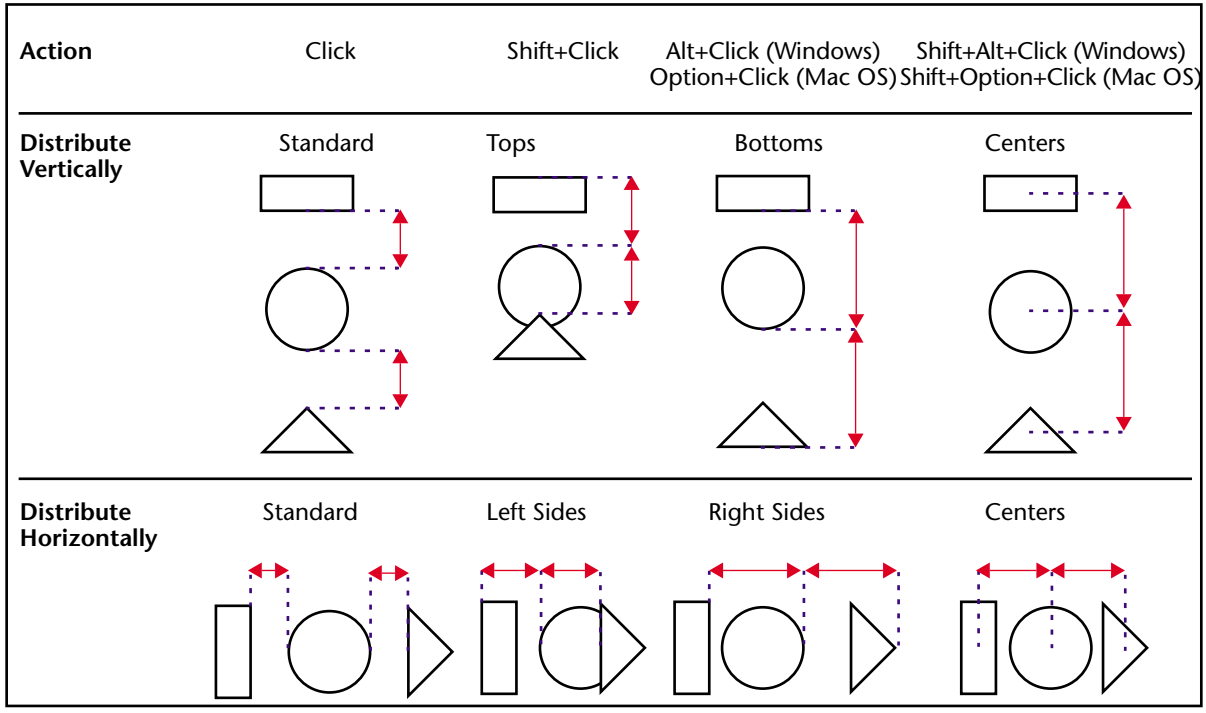
Distributing Objects

The toolbar includes two tools that let you distribute three or more objects evenly.

When you use either tool, you can modify its action by holding down the **Shift** or **Alt** keys when you click either tool.

- **Clicking**: Distributes the objects from their adjacent sides,
- **Shift+Clicking**: Distributes the objects from their left sides (horizontal) or tops (vertical),
- **Alt+Clicking**: Distributes the objects from their right sides (horizontal) or bottoms (vertical),
- **Shift+Alt Clicking**: Distributes the objects from their centers.

These rules are illustrated in the following diagram.



► To distribute a set of objects:

1 **Select the objects that you want to distribute.**

You must select at least three objects. Hold down **Shift** and click to select several objects.

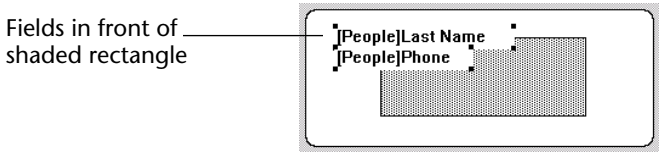
2 **If desired, hold down a modification key or key combination and click the Distribute Horizontally or Distribute Vertically tool.**

4D distributes the selected objects according to the rules you selected.

Layering Objects

You may want to create a design that uses objects in different layers. For example, you may want to place a shaded rectangle behind the fields on a label. The Label editor provides the **Move to Back** and **Move to Front** tools that let you layer objects on the label.

The following illustration shows objects in front of other objects:



Clicking the **Move to Front** or **Move to Back** tools moves the selected objects to the top or bottom layer. If you want to move the object only one layer toward the front or back, hold down the **Shift** key when you click **Move to Front** or **Move to Back**.

► To move an object to the front or back:

1 **Select the object or objects that you want to move to the back.**
Hold down **Shift** and click to select several objects.

2 **Click the Move to Front or Move to Back tool in the toolbar.**
4D moves the selected object or objects to the front or behind all the other objects.

Note When you move an object to the back, it may be hidden by objects in front of it. To see the object, select the object in front and send it to the back.

Duplicating Objects

You can duplicate any object in the label. Copies of active objects retain all the properties of the original, including foreground and background colors and fill patterns, text attributes, and display format.

► To duplicate an object:

1 **Select one or more objects.**
Hold down **Shift** and click to select several objects.

2 **Click the Duplicate tool in the toolbar.**
4D duplicates the selected object or objects.

Moving Objects

You can move objects by selecting them and dragging with the mouse. You can also use the arrow keys to move the object either one or ten pixels at a time.

- To move an object one pixel at a time, select the object and press an arrow key.
- To move an object ten pixels at a time, select the object, hold down the **Ctrl** key (on Windows) or **Command** key (on Macintosh), and press an arrow key.

Resizing Objects

You can resize any object on the label by selecting it and dragging a selection handle.

► To resize an object by dragging:

1 **Select the object you want to resize.**

2 **Move the pointer over one of the four handles that appear on the selected object.**

The pointer changes into a multi-directional arrow and the handles disappear.

3 **Drag the handle toward the center of the object to shrink it.**
OR
Drag the handle away from the object's center to enlarge it.

4D resizes the object.

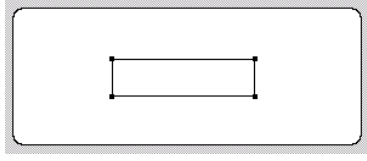
Adding a Border to an Object

You can add a one-pixel border to an object. The border can be from one to nine pixels from the object.

► To add a border:

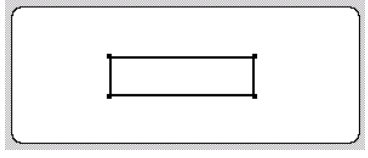
1 **Select the object.**

The selected object is indicated by selection handles.



2 **Hold down the Ctrl key (Command key on Macintosh) and press a number from 1 to 9 on the numeric keypad.**

A border is added to the object. The distance from the object (in pixels) is controlled by the number you pressed. The following illustration shows the results of pressing **Ctrl+1**.



Adding Foreground or Background Colors

4D lets you add colors to objects for display on a color monitor or (if your printer supports color) for color printing. By combining selected colors with fill patterns, you can display thousands of different color shadings.

Note Colors appear black and white on a black and white monitor. They appear as shades of gray on a gray-scale monitor. The color palettes display shades on a gray-scale monitor.

You can specify different colors for foreground pixels (pixels that appear black on a black-and-white monitor) and background pixels (pixels that appear white on a black-and-white monitor). If the object is a field or static text, the foreground color controls the color of the text and the background color controls the color of the object's rectangle.

You set foreground and background colors using the **Background** and **Foreground** picture menus in the Object Look area of the Label editor.

If your monitor supports 16 colors, choose the colors from the first 16 colors on each palette. If your monitor supports 256 colors (or more), any colors you choose will display properly.

Setting Fill Patterns

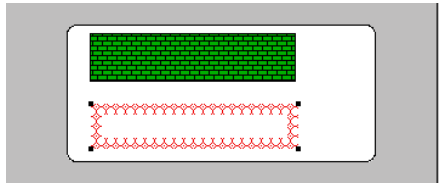
You can apply a fill pattern to any two-dimensional graphic object in the form such as an oval, a rectangle, a line, the enclosed area of a field or static text object, and a two-dimensional object's border.

The **Fill** picture menu controls the fill pattern for the selected object. The **Border** picture menu controls the fill pattern for the border.

Setting Border Patterns

You can set patterns for the borders of any object in the form that has a border such as an oval, a rectangle, and a grid object. The border patterns available are the same as the fill patterns. The appearance of the border also depends on the line width you have specified for the border.

The following illustration shows a fill pattern applied to the upper rectangle and a border fill pattern applied to the bottom rectangle.



Setting Line Width

The **Line Width** pop-up menu controls the width of a line or the width of the border of a two-dimensional object.

To set a line width, select the object and choose a line width from the **Line Width** pop-up menu.

Restoring the Default Look

When you add an object to the label, it takes on the Object Look attributes of the most recently created object. If you want to assign the default Object Look attributes to this object (or any object), select the object and click **Default Look**. The foreground and background colors, fill patterns, and border width are reset to the default settings.

Adding a Graphic to the Label

In addition to the drawing tools in the toolbar, you can paste a graphic from the Clipboard into the label.

► To paste a graphic into the label:

1 **Place the graphic on the Clipboard.**

2 **Choose Paste from the 4D Edit menu or press Ctrl+V (Command–V on Macintosh).**

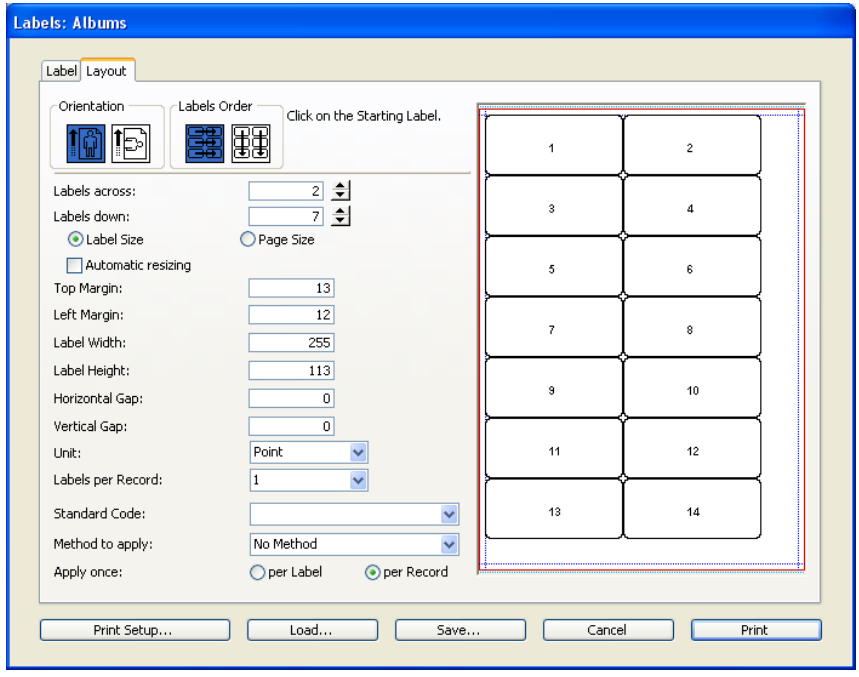
The graphic appears in the Label Preview area with selection handles. You can then move, align, distribute, or resize the object like any other object.

Deleting Objects

You can delete the selected object by pressing the **Backspace** key (**Delete** key on Macintosh). If the selected object contains concatenated fields, the last field is removed from the object. If you want to delete the entire object, continue pressing the **Backspace** or **Delete** key.

Specifying the Label Layout

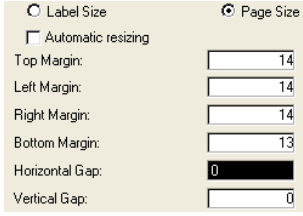
You specify the label layout with the Layout page of the Label editor. You can display the Layout page at any time by clicking its tab.



The Layout page contains the following elements:

- **Orientation and Labels Order buttons**: These buttons enable you to specify the page orientation and the order in which information is assigned to labels. Keep in mind that the orientation of the *sheet* is independent from that of the *page*. If you modify this parameter, be sure to change the configuration of the page accordingly in the standard print setup dialog box.
- **Labels across and down boxes**: These boxes are used to control the size of labels by specifying the number of labels that appear on your label paper.

- **Layout preview area**: This area provides a reduced view of how an entire page of labels will look, based on the dimensions you enter in the Label editor. The page preview also reflects the paper size selected in the Print Setup dialog box. You can also select the first label on the page to be printed. The red border indicates the size of the physical page and the blue border indicates the size of the printable area.
- **Label Size and Page Size radio buttons**: These buttons are used to select the label or the page for setting label and page dimensions. If you click Label Size, you can enter the label width and label height in the appropriate areas. If you click Page Size, you can enter values for right margin and bottom margin, as shown below.



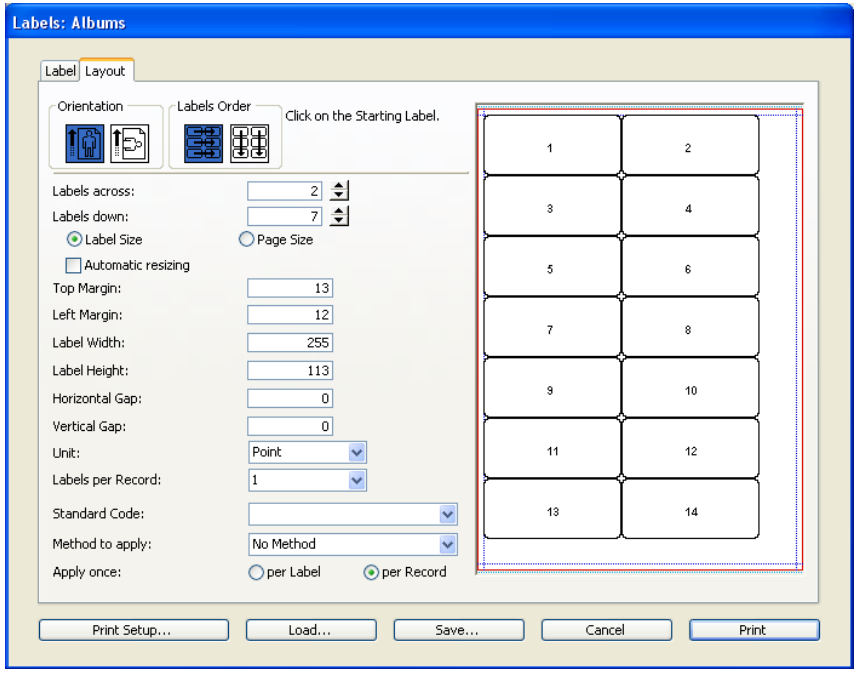
- **Margin boxes**: These boxes are used to specify the dimensions of the label and the page size, depending on the radio button you select. After you have entered the margins of your label paper, you may need to make some additional adjustments so that the label text is centered in the labels. You can use both positive and negative numbers in the Margin boxes to increase and decrease the margins.
- **Automatic Resizing**: If **Automatic Resizing** is checked, the values in the Label Width and Label Height entry areas are set automatically.
- **Horizontal Gap**: This area controls the amount of space between label columns.
- **Vertical Gap**: This area controls the amount of space between label rows.
- **Unit drop-down list**: This drop-down list allows you to change the units in which you specify your label and label page measurements. You can use pixels, millimeters, centimeters, or inches.

- **Labels per record drop-down list**: This control lets you print more than one copy of each label. If you print more than one copy, 4D prints the copies consecutively rather than making copies of the label pages.
- **Standard Code drop-down list**: This control lets you specify the label, page dimensions, and margins by choosing a standard commercial label paper from the drop-down list.
- **Method to apply**: This control lets you choose a method that will be run at print time. For example, you can execute a method that posts the date and time that each label was printed.
- **Apply Once radio buttons**: These radio buttons are used to specify whether to run the method once per label or once per record. This control has meaning only if you are printing more than one copy of each label and you are also executing a method at print time.
- **File buttons**: These buttons provide options for page setup, printing, saving, and loading label designs.

► To specify the layout of the label sheet you are using:

1 Click the Layout page tab.

The Layout page of the Label editor appears.



You can specify the design of the label sheet using the entry areas on the Layout page or choose a standard design from the Standard Code drop-down list. This drop-down list contains specifications for a wide variety of standard commercial label sheets.

2 **Click the Print Setup button.**

The Print Setup dialog box for your operating system appears.

3 **Choose the desired printer and click OK.**

If necessary, the Label Preview area changes to reflect your selection.

4 **If appropriate, choose the type of label paper you are using from the Standard Code drop-down list.**

The remaining entry areas on the page change to reflect the selected label paper's characteristics. If necessary, you can modify these specifications.

5 **Click the appropriate Orientation and Labels Order radio pictures.**

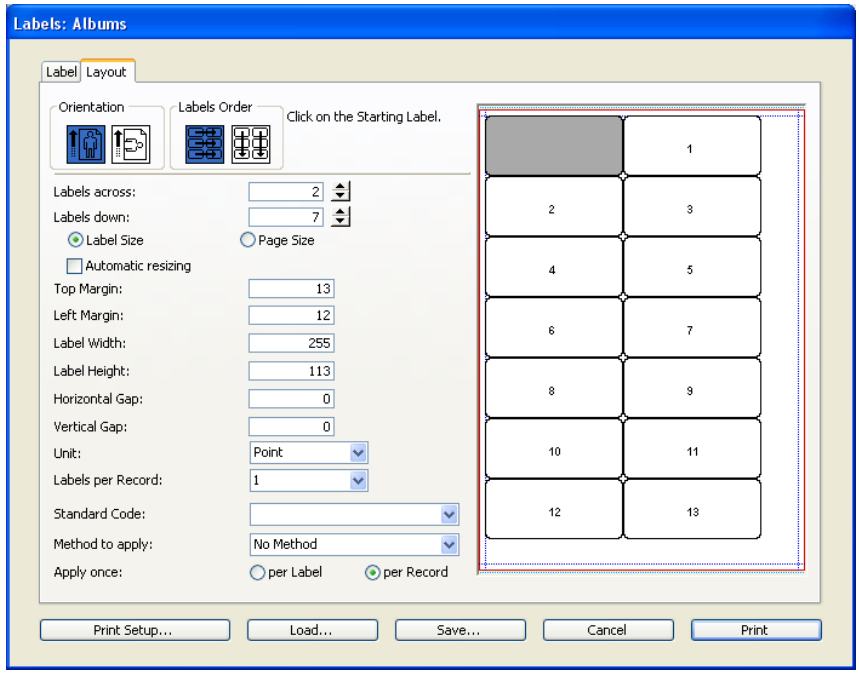
You can choose between portrait and landscape orientation and horizontal or vertical order. Remember that the orientation of the *sheet* is independent from that of the *page*. If you modify this parameter, be sure to change the configuration of the page in the standard print setup dialog box.

6 **Enter the number of labels in each row of your label sheet in the Labels Across box and the number of labels in each column in the Labels Down box.**

The Label Preview area adjusts to display the appearance of the labels on a printed page.

7 **If the first sheet of label paper is partially used, click on the first blank label in the Label Preview area.**

4D will begin printing labels on the label you indicated. The following illustration shows the Preview area after clicking on the second label area.



8 If desired, choose a unit of measurement from the Unit drop-down list to use for entering margin sizes.

9 Enter values to reflect the margins on your label paper.

Use the **Label Size** and **Page Size** radio buttons to control whether you use the entry area for the size of the label or the size of the page.

The size of the individual labels in the label page preview will adjust to accommodate the margins. For example, if you increase the size of your margins to two inches, top and bottom, the size of the individual labels will shrink to maintain the same number of labels that you specified earlier.

Because some printers use portions of the margin to hold the label sheet in place, the printer may not take the full margins into account when printing your labels. In this case, you may need to adjust the margin settings so the label text is properly centered in each label.

Since the printer uses part of the margins, the printer begins measuring the margins from a point that is not precisely at the edge of the label paper. When the labels are printed, the label text may appear skewed to the right or to the bottom of the label paper.

To compensate for this lost margin space, you can use negative numbers in some of the margin boxes. When placed in the Right margin box, negative numbers pull the label text to the right. When entered in the Top margin box, negative numbers pull the label text toward the top of the page.

Note As a rule, using a negative number in a margin box moves the label text toward the margin you are setting.

10 **(Optional) If you want to print more than one copy of each label, use the Labels per Record drop-down list to choose the number of copies to print.**

The copies are printed consecutively on the label paper. 4D does not duplicate the entire label page.

11 **(Optional) If you want to run a method when the labels are printed, choose the method from the Method to Apply drop-down list.**

12 **(Optional) If you are running a method and printing more than one copy of each label, click either the Once Per Record or Once Per Label radio button in the Apply Once area.**

This control has no effect unless you are using both the multiple copies and method features.

Saving and Loading Label Designs

4D lets you save each label design as a file that you can open from the Label editor. By saving label designs, you can maintain a library of labels that you can use according to your needs.
Label designs store the parameters set on the Label and Layout pages.

Saving a Label Design

To save a label design:

1 **Click the Save button.**

4D displays a dialog box where you can enter a file name for the label design.

Note Label files on Windows are denoted by the file extension .4LB.

2 **Enter a filename for the label design and click the Save button.**

Loading a Label Design

You can load the label design whenever the Label editor is active.

▶ To load a label design:

1 Click the Load button.

4D displays an open-file dialog box where you can select the filename of a label design. Double-click the filename or select the filename and click **Open**.

4D replaces the current label design with the design you selected.

Printing Labels

After you have completed your label design, you can preview or print the labels. You may want to print first on regular paper so that you can check the placement of text before you use the more expensive label paper.

▶ To print your labels:

1 Click the Print button.

The Print dialog box appears for the printer you selected in the Print Manager. If you check the **Print Preview** check box before clicking **OK**, the labels are previewed on screen.

If you are printing using a form, 4D will use the selected form to print the labels. 4D assumes that the form was designed to print labels.

After the labels are printed, 4D closes the Label editor and returns you to the form you were using when you opened the Label editor.

21 Graphs

4D allows you to create a wide variety of two- and three-dimensional plots without having to export the data to a graphics package. You can create graphs from the data in your database or from data that has been copied to the Clipboard from another application. You can graph data directly from fields or you can graph the results of calculations on data. You create graphs in 4D using the built-in 4D Chart plug-in.

Because graphing capability is fully integrated into 4D, you can graph data in your database and update your graphs when the information in the database changes. You can print your graphs or copy them to the Clipboard and paste them into other applications.

Finally, 4D Chart adds over 100 commands to the 4D language which enable you to control tasks that you normally perform manually. For example, you can use 4D Chart commands to create new graphs, modify graph features, open and save documents, and execute any 4D Chart menu command. For information on these commands as well as on elements specific to the use of 4D Chart in the Application environment, please refer to the *4D Chart Language Reference* manual.

Note It is also possible to generate graphs in 4D by using the integrated SVG rendering engine. This function is only available by programming via the GRAPH command. For more information, please refer to the 4D *Language Reference* manual.

Managing 4D Chart Documents and Windows

4D Chart documents can be created in plug-in areas on forms or in separate plug-in windows. This section explains how to create, open, and save 4D Chart documents in both types of areas.

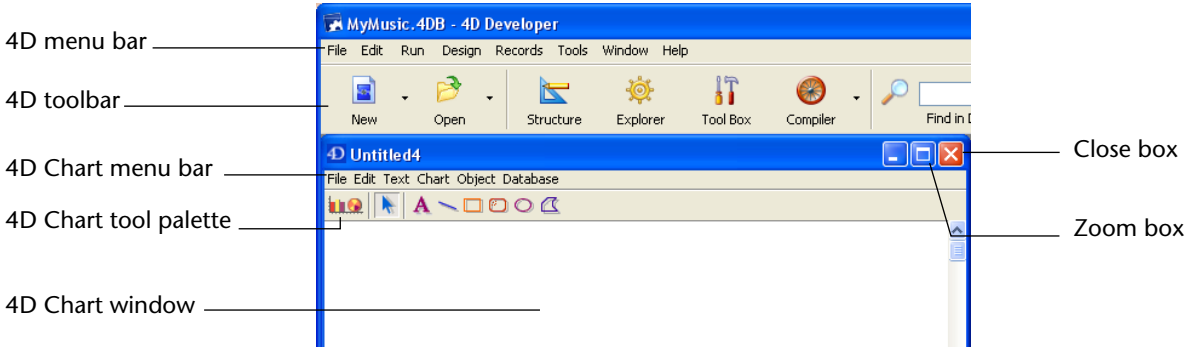
This section explains the basics of managing 4D Chart documents, including:

- Using 4D Chart in a plug-in window
- Using 4D Chart in a 4D form
- Hiding and showing 4D Chart features
- Creating a new document
- Opening an existing document
- Saving a document
- Setting the document size.

Using 4D Chart in a Plug-in Window

You can use 4D Chart in its own window. When used in its own window, 4D Chart operates as if it were a separate application.

When you open 4D Chart in a plug-in window, the window has its own menu bar. 4D's menu bar remains at the top of the screen.



When you expand the window to full screen size by clicking the window's zoom box, 4D's menu bar remains at the top of the screen and 4D Chart's menu bar remains within the 4D Chart window.

To generate and modify a graph in an external 4D Chart window, you have two possibilities:

- Call the Chart Wizard directly from 4D. After it has been set up, the graph is displayed in a new external window,
- Open a blank 4D Chart window then use the menus and functions of the plug-in to create your graph.

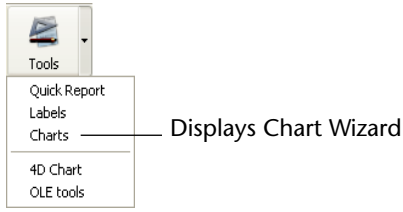
Opening an External Window Displaying the Graph Directly

When you create a graph from the Design mode of 4D, an external window is automatically opened to display it.

► To create an external window displaying the graph directly:

1 **Make sure that the current selection contains the records that you want to represent graphically.**

2 **Choose Charts... from menu associated with the "Tools" button in the 4D tool bar.**



The Chart Wizard appears. The use of this wizard is described in the "Creating a Graph" paragraph on page 1076.

3 **Set the parameters of your graph and click on OK.**

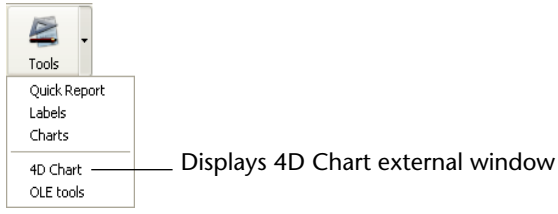
An external 4D Chart window is created to display the graph.

Opening a Blank 4D Chart External Window

You work with external windows in the Design mode of 4D.

► To open 4D Chart in an external window:

1 **Choose 4D Chart from the Tools menu or from the menu associated with the "Tools" button in the 4D tool bar.**



A new 4D Chart document opens in an external window.

You can open additional 4D Chart windows by choosing 4D Chart again from the **Tools** menu. Opening several 4D Chart documents at the same time allows you to compare documents, copy and paste between documents, and move from one document to another simply by clicking on the appropriate window.

The titles of all 4D Chart windows appear at the bottom of the Window menu. You can bring any document to the front by choosing it from this menu.

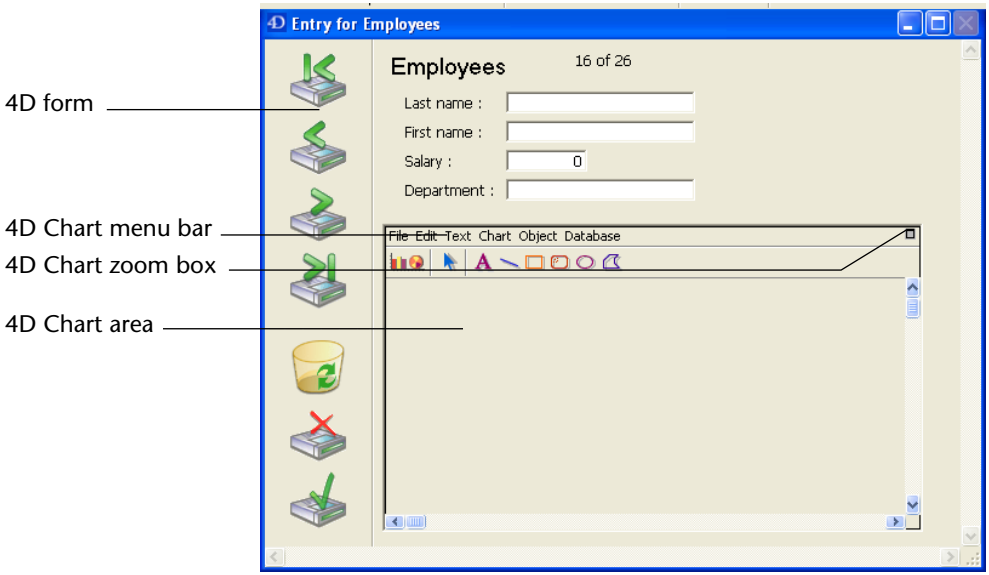
When you save a document, its title bar displays the document name. Saved documents automatically receive the extension ".4CT".

You can close a plug-in window at any time by clicking the window's control-menu box (on Windows) or the close box (on Macintosh).

Using 4D Chart in a Form

You can place a 4D Chart area in any form. If you use a table form, you can save the contents of the 4D Chart area with each record. When you use 4D Chart in an input form, you can display a graph for each record in the table. You can also use a 4D Chart area in an output form.

When 4D Chart is used in a form, the 4D Chart menu bar appears at the top of the 4D Chart area. You can choose menu commands from either the 4D menu bar or the 4D Chart menu bar.



► To expand a 4D Chart area:

1 **Choose Go to Full Window from the 4D Chart File menu.**
OR
Click the 4D Chart area's zoom box.

The document expands to fill the entire screen, and 4D Chart's menu bar temporarily replaces 4D's menu bar.

The expanded window has a size box and close box.

When the document window is expanded, the **Go to Full Window** menu item in the File menu changes to **Return to Form**.

► To reduce the window and return to the form:

1 **Choose Return to Form from the File menu.**
OR
Click the close box.

Creating a 4D Chart Area in a Form

You can add a chart area in an input form and save a graph with each record in the table.

For more information about inserting a plug-in area into a form, refer to the "Plug-in Areas" paragraph on page 592.

Minimum Size of the 4D Chart Area

By default, if the size of the 4D Chart area included in the form is less than the minimum size (width 300 pixels, height 150 pixels), it will appear as a button. The button title is the name of the variable associated with the plug-in area.

When the user clicks on this button, the area is displayed in full-page mode.

It is possible to disable this operation using the CT SET ENTERABLE language command (the area is then never displayed as a button).

Hiding and Displaying 4D Chart Features

You can choose to hide or display several of 4D Chart's features, including:

- The 4D Chart menu bar
- The Object Tool palette
- The Chart Tool palette
- Scroll bars
- Rulers.

Use the **Display** submenu of the **Edit** menu of the plug-in area to hide or display these items. Items that are checked in the Display submenu are displayed in 4D Chart.

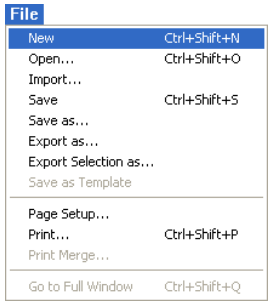
Note If you have hidden the 4D Chart menu bar, you can display it by pressing **Ctrl-Shift-M** on Windows or **Command-Shift-M** on Macintosh.

Creating a New 4D Chart Document

You can create a new, blank 4D Chart document at any time. The new document replaces the current document. If you have made changes to the current document, you will be prompted to save them before opening a new document.

► To create a new document:

1 **Choose New from the 4D Chart File menu.**



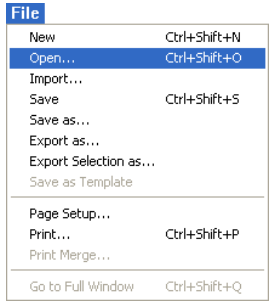
An empty document appears in which you can create a new graph.

Opening a 4D Chart Document

You can open any previously saved 4D Chart document in a 4D Chart area. The newly opened document replaces any current document.

► To open a 4D Chart document:

1 **Choose Open from the 4D Chart File menu.**



A standard open-file dialog box appears.

2 **Select a document.**

4D Chart documents have the extension ".4CT".

3 **Click Open.**

The document opens in the current 4D Chart window.

Note You can also open a PICT file in your graph. This point is covered in the "Importing a PICT Document in a 4D Chart Window" paragraph on page 1062.

Saving a 4D Chart Document

You can save the contents of a 4D Chart document regardless of whether the document was created in a plug-in window or in a form. 4D Chart offers several ways to save documents:

- As a file
- As part of a record
- As a template for a 4D Chart area.

You can also save a selection of objects, for instance a graph, as a PICT document (Macintosh format). In this case, the document saved is a picture that can no longer be modified. This point is covered in the "Exporting a 4D Chart Document in PICT Form" paragraph on page 1062.

Saving a 4D Chart Document as a File

Any document that you create using 4D Chart can be saved as a separate document for use in another place — either in the same database, in another database that uses 4D Chart, or in an entirely different application. You use the standard **Save** and **Save As** menu items in the 4D Chart **File** menu to save and update individual documents.

Documents you save can be opened with the **Open** menu item in the 4D Chart **File** menu. It makes no difference whether a document is created in a plug-in window or in a form; a document can be saved and opened in either place.

If you save a 4D Chart document as a file, when you reload the file, the document appears exactly as it was when you saved it. To update the information in any graphs in the document, you must use the **Update** menu item in the **Chart** menu, which is described in more detail in the "Updating the Data in a Graph from the Database" section on page 1084.

► To save a 4D Chart document:

1 **Choose Save As from the 4D Chart File menu.**
Notice that you choose **Save As** from the 4D Chart **File** menu, not the 4D **File** menu. 4D Chart displays a save-file dialog box.

2 **Enter a filename for the document.**

3 **Click Save.**
4D Chart saves the document with the filename you entered.

Note 4D Chart documents have the extension ".4CT".

Saving a 4D Chart Document as Part of a Record

If you have created a 4D Chart plug-in area in a form and want its contents to be saved automatically with each record, you need to create a BLOB field in the table to which the form belongs in order to store the contents of the area. The contents of the area are then saved automatically with each record when it is validated.

Note This mechanism cannot be used with project forms.

► To save a 4D Chart document as part of a record:

1 **Add a BLOB field to the table whose form contains the 4D Chart area.**
For more information about creating a field, refer to the "Creating a Field" paragraph on page 265.

2 **Give the field the same name as the external area that you have created on the form and add an underline (_) to it.**
For example, if your external area is named MyArea, the field must be named MyArea_.
Each graph is thus saved as a part of the record.

If you save a 4D Chart document as part of a record, when you reload the record, the document appears exactly as it was when you saved it. To update the information in any graphs in the document, you must use the **Update** menu item, which is described in more detail in the "Updating the Data in a Graph from the Database" section on page 1084.

Saving a 4D Chart Document as a Template

If you have inserted a 4D Chart area into a form, you can create a standard document that is the same for every record by saving the document as a template. It is not possible to use templates with external windows or with project forms.

When you save a document as a template, the template is used by default for every new record that is opened in the table form.

Each document starts with the same template, but any unique modifications you make to it are saved with the record. There can be only one template for each 4D Chart area on a form.

If there is a graph in the document that is saved as a template, the graph is updated automatically for each record, if possible. 4D Chart can update a graph only if it was created from data stored in records in the database.

► To save a document as a template:

1 Choose Save as Template from the 4D Chart File menu.

4D Chart saves the document with a special filename. The filename is the name of the 4D Chart area on the form, plus an underscore and the extension .4CT.

For example, an area named "Document" will have a template file named "Document_.4CT".

Because 4D Chart automatically names the file and places it in the database directory; no save-file dialog box is displayed.

4D Server By default, 4D Chart templates are read from and saved to the client machines. You can use the CT SET AREA PROPERTY command of the 4D Chart language to specify that templates be read from and saved to the server machine.

If you make changes to the document and want to incorporate those changes in the template, choose **Save as Template** again.

You can create a template for a 4D Chart area without using the **Save as Template** menu item simply by saving the document with a filename that is the same as the name of the 4D Chart area, plus an underscore and the extension .4CT, and placing the file in the database directory. This feature allows you, for example, to create a document in one database and use it as a template in another database. You can also create a document in one document area and use it as a template in another document area.

You can temporarily disable the use of a particular template file by changing the template filename or moving the file out of the database directory.

Importing a PICT Document in a 4D Chart Window

You can open documents of the PICT type in a 4D Chart window. These documents can come from different sources (export of a 4D Chart graph in PICT form, design software, etc.). The imported document behaves like a simple object; you cannot modify its attributes.

► To import a PICT document:

1 **Choose the Import... command in the File menu of 4D Chart.**
A standard open file dialog box appears.

2 **Select the PICT file you want to open (*.PCT extension under Windows) and click Open.**
If the imported file is valid, its contents are displayed in the 4D Chart window.

Exporting a 4D Chart Document in PICT Form

You can export a 4D Chart document or a selection of objects in it as a PICT file. In both cases, the objects do not keep their unique 4D Chart properties; the exported document is a static picture of the objects in the 4D Chart area or in the external window. You can import it as a PICT in 4D Chart or any other application that opens PICT files.

► To export a 4D Chart document in PICT form:

1 **Choose the Export as... command in the File menu of 4D Chart.**

2 **Select a name and location for your file in the standard dialog box then click Save.**

► To export a selection of 4D Chart objects in PICT form:

1 **Select the object(s) that you want to export in PICT form.**

2 **Choose the Export Selection as... command in the File menu of 4D Chart.**
A standard save file dialog box appears.

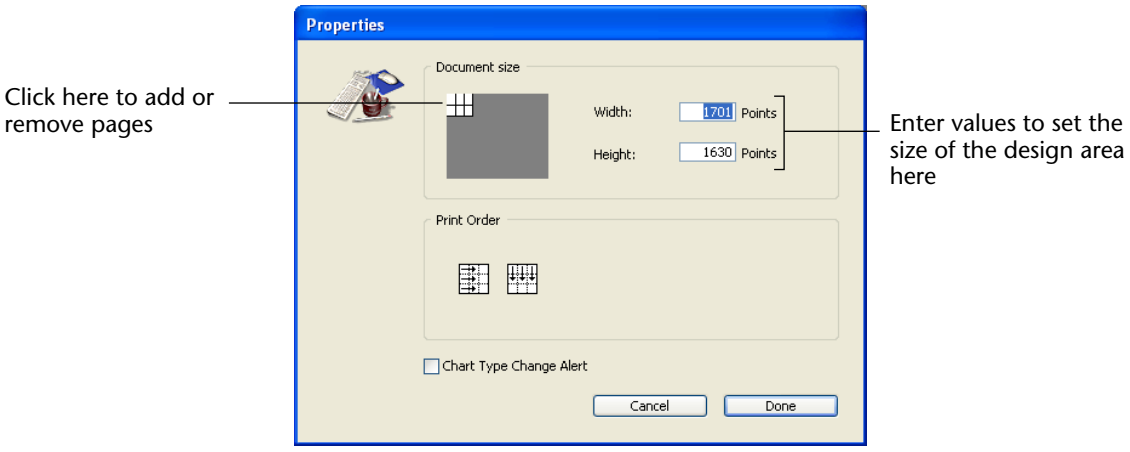
3 **Select a name and location for the file and click Save.**
Whether you export a 4D Chart document or a selection of objects, a PICT file (*.PCT extension under Windows) is created on the disk.

Setting the Document Size

You can change the document size by changing the number of pages in the document or by specifying a new document size in points. The maximum drawing size is 3500 x 3500 points. You can add pages to the right of the first page, below it, or both.

► To change the document size:

1 **Choose Properties from the 4D Chart Edit menu.**
The following dialog box appears.



The document size is indicated by the point values to the right of the model document area and by the white rectangles in the model document area.

2 **Click in the Document size area to adjust the number of pages in the drawing.**
OR
Enter the exact size of the document (in points) in the Width and Height text areas.

Choosing a Graph Type

4D Chart allows you to create two-dimensional and three-dimensional graphs from within your database. Once you decide what data you want to graph, you should think about what type of graph will best display that data.

The nature of the data that you are graphing will help indicate the type of graph you should choose. For example, a line graph is best suited for showing how values (such as “number of units sold”) change over time.

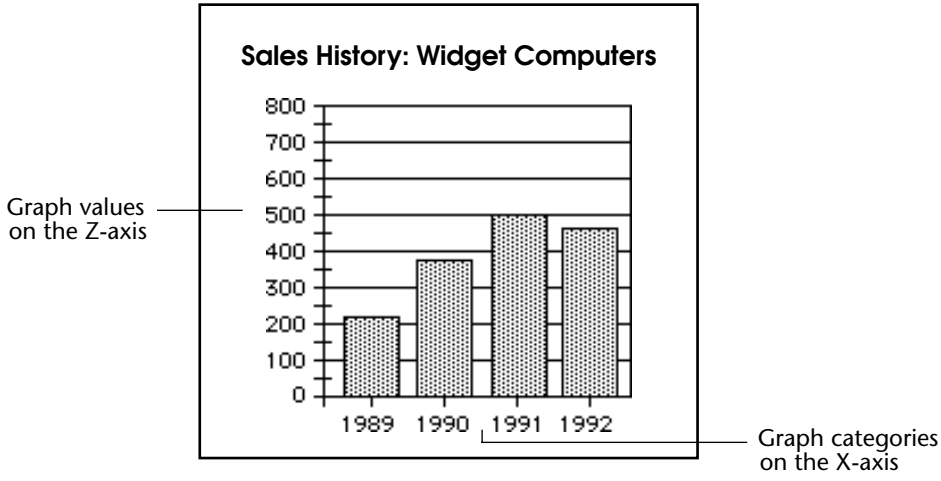
Don’t be afraid to experiment with different graph types. Once you have created a graph, you can easily convert it to another graph type. Creating graphs and changing their types are discussed in the “Creating a Graph” section on page 1076.

This section discusses the types of two- and three-dimensional graphs that you can create using 4D Chart. The features of each type of graph are described, along with the type of data that the graph is best suited to show.

Choosing a Two-dimensional Graph Type

This section briefly describes the parts of a two-dimensional graph and then presents each two-dimensional graph type.

The following diagram shows a two-dimensional (two-axis) graph:

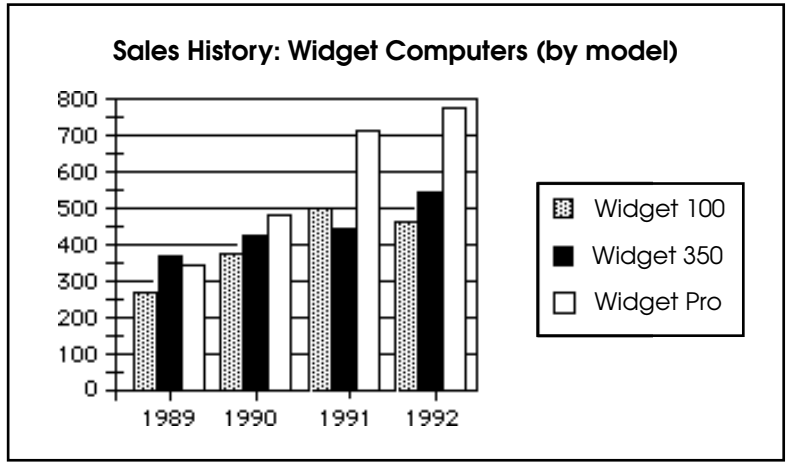


The X-axis is also called the *Category axis*. The X-axis displays the categories into which the information is divided. For instance, if you are graphing the number of computers sold per year, you would place the years on the X-axis.

The Z-axis is also called the *Values axis*. You use the Z-axis to display the values calculated for each category. In the computer sales example, the Z-axis would contain the values for the numbers of computers sold per year. For instance, if there were 500 computers sold in 1991, the Z-axis value corresponding to the X-axis category “1991” would be 500.

Another concept in graphing is the *data series* (or simply, *series*). Each category is composed of one or more series that further breaks down the information displayed in the graph. In this example, there was only one series, “computers sold.” Therefore, the series is not represented by a separate field or formula.

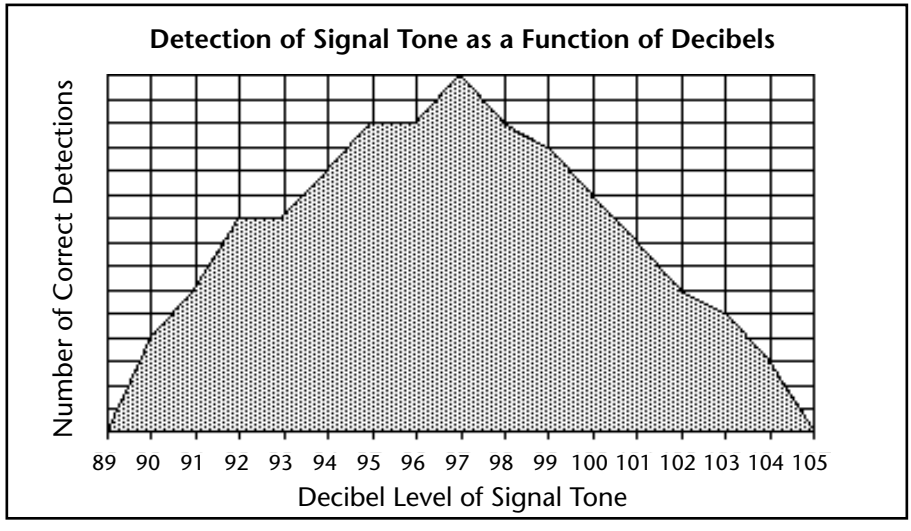
However, suppose you want to compare the sales volumes from 1989 to 1992 for three models of computers. In this case, each computer model becomes a distinct series on the graph. All series share the same categories (in this case, the years 1989 through 1992), but have their own values.



Another way to graph this data would be to create a three-dimensional graph of the data. This type of graph is discussed in the "Choosing a Three-dimensional Graph Type" section on page 1071.

Area Graphs

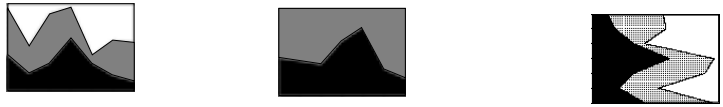
Area graphs are commonly used to show the magnitude of values over time, but can show values over any continuous category.



The categories on the X-axis should be continuous, such as times or temperatures. Discrete categories, such as salespeople or products, are better suited to column graphs.

Options
The following options are available for Area graphs:

- **Stacked**: When graphing multiple series, stack the areas for the series.
- **Stacked, proportional**: When graphing multiple, stacked series, show the series as proportions of a 100% whole.
- **Horizontal**: Make the X-axis the vertical axis and the Z-axis the horizontal axis.

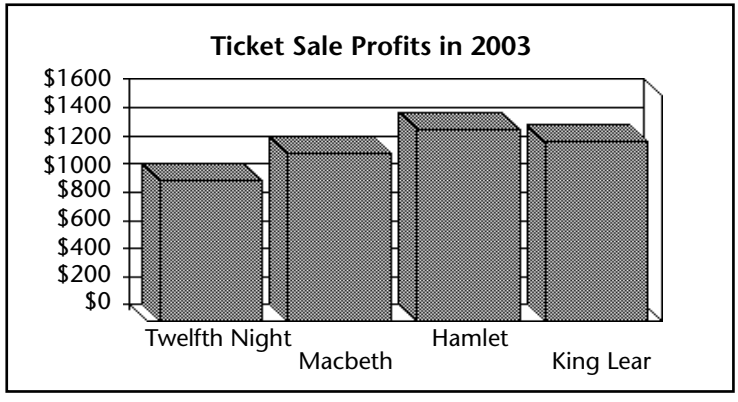


Stacked | Stacked, proportional | Horizontal

Column Graphs

Column and bar graphs are the most frequently used graph types for business data. They are usually used to compare one item to another, or one or more items over a period of time.

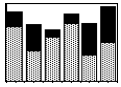
The following column graph uses the Depth feature, which makes each column look three-dimensional.



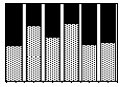
Note The Depth feature is discussed in more detail in the "Modifying Depth in a Two-dimensional Graph" section on page 1106.

Options

The following options are available for Column graphs:



- **Stacked**: When graphing multiple series, stack the columns for the series in each category.

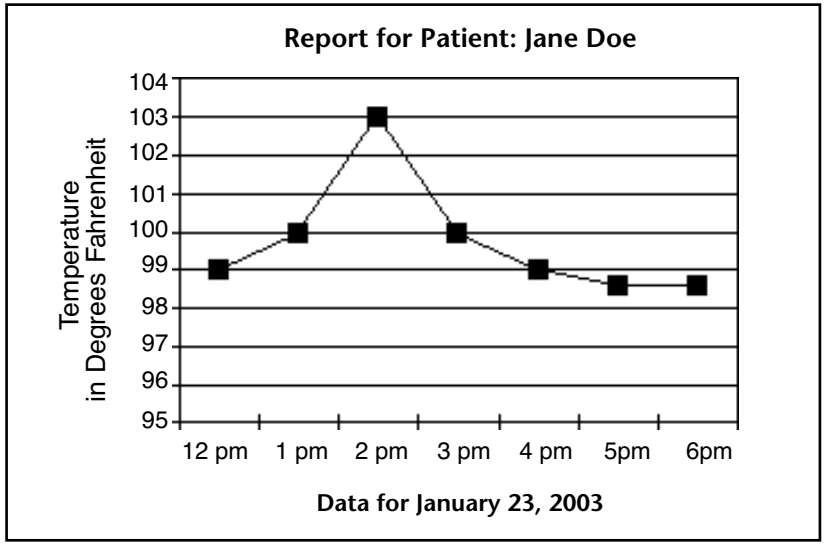


- **Stacked, proportional**: When graphing multiple, stacked series, show the series as proportions of a 100% whole.

- **Horizontal**: Make the X-axis the vertical axis and the Z-axis the horizontal axis. This option creates a bar graph. Bar graphs show the categories on the vertical axis, with the values expressed by the length of the bars. Bar graphs are often a useful alternative to column graphs, especially when the category labels are long.
- **Overlap**: When graphing multiple series, use this option to specify the percentage by which the series columns within each category overlap each other.
- **Gap Width**: Use this option to specify the spacial gap between the columns from one category to the next. The larger the gap, the narrower the columns for each category.

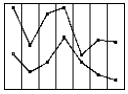
Line and/or Scatter Graphs

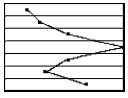
Line graphs are often used to show the rate of change of values over time. Scatter graphs can show relationships and trends in your data. Such graphs are most often used when there are a large number of different values of the X-axis field, as in a scientific study.



The values on the X-axis should be continuous or ordered, such as times or temperatures. Discrete categories or categories that do not have an order — such as salespeople or products — are better suited to column graphs.

Options
The following options are available for Line and/or Scatter graphs:



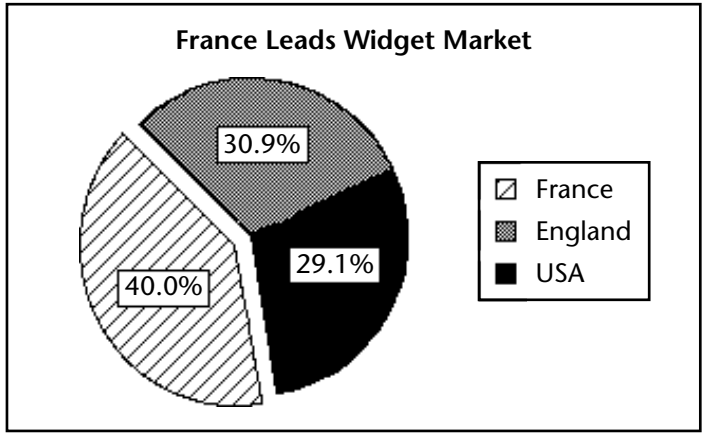


- **Stacked**: When graphing multiple series, stack the lines for the series in each category.
- **Horizontal**: Make the X-axis the vertical axis and the Z-axis the horizontal axis.
- **Displaying lines and/or points**
  You can:
  - **Show Points**: Each value appears as a dot in the graph.
  - **Show Lines**: Only lines appear in the graph.
  - **Show Both**: Lines and Points appear in the graph.

Pie Charts

Pie charts show data as a percentage of a whole. Your data does not need to be expressed as percentages; 4D Chart automatically converts the data into percentages when it creates the pie chart.

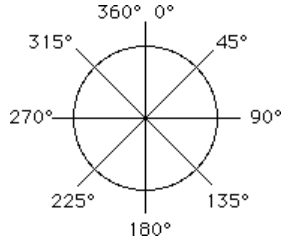
Each pie chart can have only one series. Categories are displayed in the legend.



Options

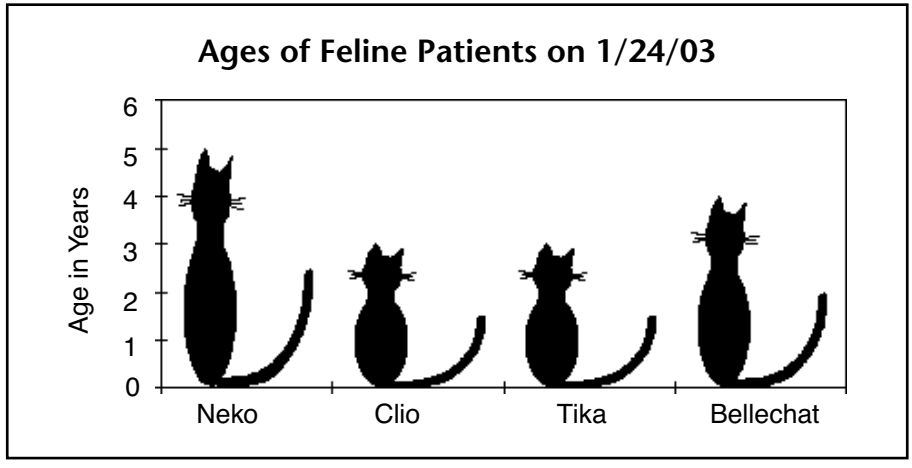
The following option is available for pie charts:

- **Start Angle**: Use this option to specify the rotation of the chart. The specified angle determines the placement of the first edge of the first category in the chart.



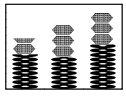
Picture Graphs

Picture graphs are similar to column graphs, except that you can substitute a picture for the column.

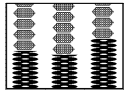


Options

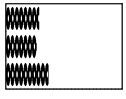
The following options are available for Picture graphs in 4D Chart:



- **Stacked**: When graphing multiple series, stack the pictures for the series in each category.



- **Stacked, proportional**: When graphing multiple, stacked series, show the series as proportions of a 100% whole.



- **Horizontal**: Make the X-axis the vertical axis and the Z-axis the horizontal axis. This option creates a bar picture graph.
- **Overlap**: When graphing multiple series, use this option to specify the percentage by which the series within each category overlap.

- **Gap Width**: Use this option to specify the spacial gap between the picture columns from one category to the next. The larger the gap, the narrower the picture columns.

  Additional options for aligning and adjusting the pictures are discussed in the "Modifying Graph Features" paragraph on page 1092.

2D XY Graphs (Scatterplots)

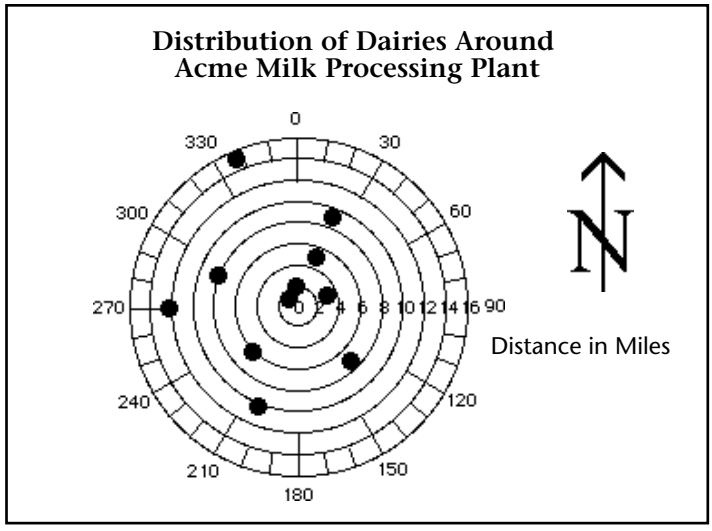
Each data point in a scatterplot represents a pair of values. The X-axis is also a value axis.

When the 2D XY type is selected, the option Category is replaced with Values (X) in the Chart>Axis, Chart>Grid Lines and Chart>Titles menus.

Options
The following options are available for 2D XY graphs in 4D Chart:

- **Plots shape**: You can display None, Circles, Squares, or Star plots.
- **Lines appearance**: You can display None, Straight, or Arrows lines.
- **Show Regression Line** (y = ax+b): Allows viewing the predictive relation between the X- and Y-axis values (if appropriate).
  There is one regression line for each series of points. Using **Ctrl+click** (Windows) or **Command+click** (Mac OS), you can select the points associated with the regression line.
- **2D XY variant: Polar charts**
  Polar charts are used to show the distribution of data around a central point. In a polar chart, each point is plotted in terms of degrees from the zero point and distance from the center.

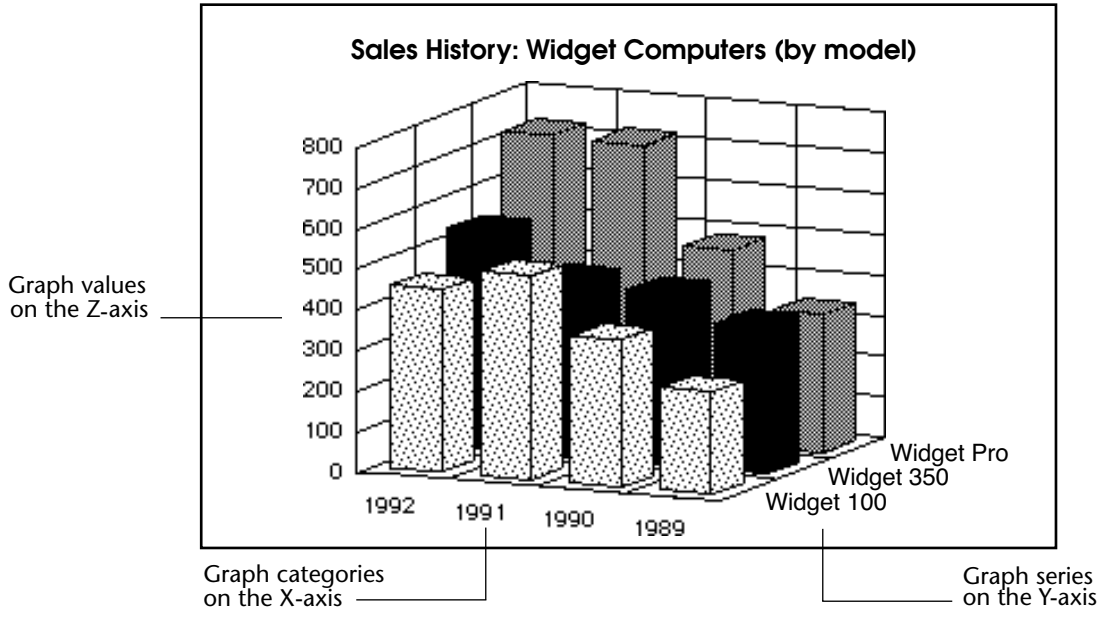
Each data point on a polar chart is plotted using one Z-axis field to specify the angle in degrees and a second Z-axis field to specify the distance from the center.



Choosing a Three-dimensional Graph Type

This section briefly describes the parts of a three-dimensional graph and then presents each three-dimensional graph type.

The following diagram shows a three-dimensional (three-axis) graph:



As with a two-dimensional graph, the X-axis (or Category axis) displays the categories into which the information is divided. For instance, if you are graphing the number of computers sold per year, you would place the years on the X-axis.

In a three-dimensional graph, the Y-axis is also called the Series axis. Each category is composed of one or more series that further breaks down the information for each value of the X-axis field. Each data point in the graph is the intersection of one category and one series. In the example illustrated above, each computer model is a series.

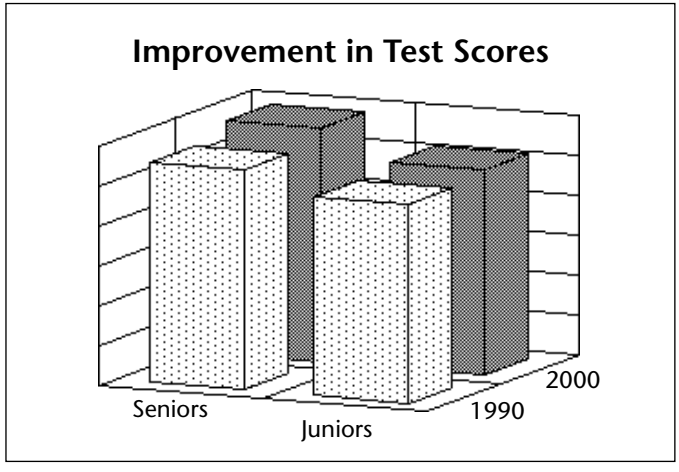
In a two-dimensional graph, the series remain on the X-axis. The columns representing each series are distinguished by their patterns. In a three-dimensional graph, the series are displayed on their own axis. Each series still has its own pattern, but now the series are also separated from each other in space.

The Z-axis displays the values calculated for each combination of category and series.

In the computer sales example, the Z-axis would contain the values for the numbers of each type of computer sold per year. For instance, if there were 725 Widget Pro computers sold in 1991, the Z-axis value corresponding to the X-axis category "1991" and the Y-axis series "Widget Pro" would be 725.

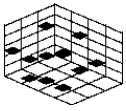
3D Column Graphs

3D Column graphs, like 2D Column graphs, compare one item to another, or one or more items over a period of time.



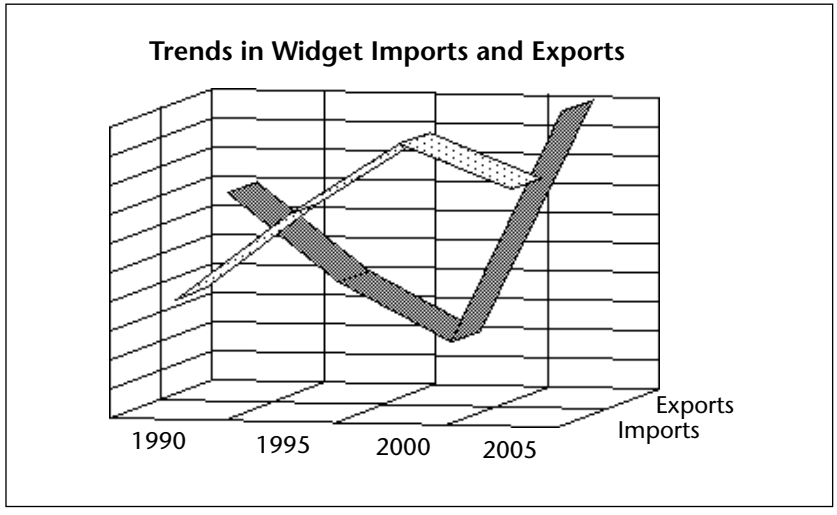
Options
The following options are available for 3D Column graphs:

- **Category Gap/Width**: The gap between or width of the categories in the graph. The larger the category gap, the smaller the category width.
- **Series Gap/Width**: The gap between or width of the series in the graph. The larger the series gap, the smaller the series width.
- **Tops only**: Show only the top side of each column.



3D Line Graphs

3D Line graphs show trends in your data. They should be used for continuous categories, such as time.

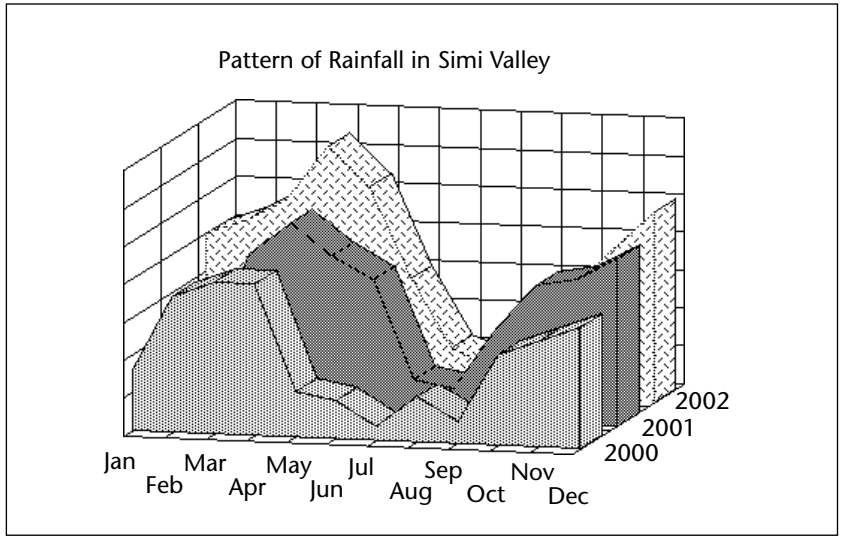


Option
The following option is available for 3D Line graphs:

- **Series Gap/Width**: The gap between or width of the series in the graph. The larger the series gap, the smaller the series width.

3D Area Graphs

3D Area graphs emphasize the volume or size of the series over a continuous category, such as time.



Option

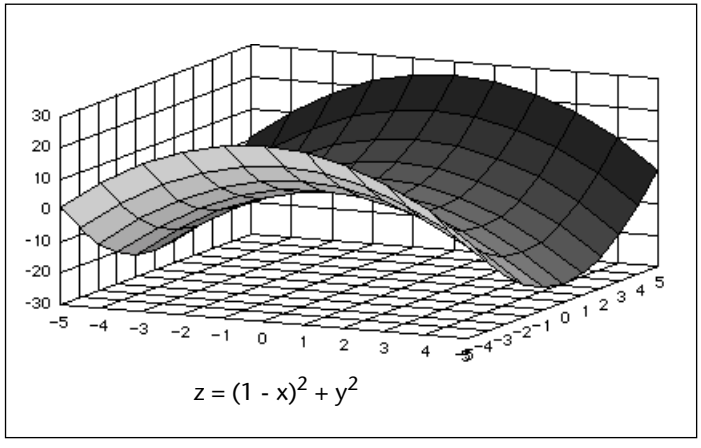
The following option is available for 3D Area graphs:

- **Series Gap/Width**: The gap between or width of the series in the graph. The larger the series gap, the smaller the series width.

3D Surface Graphs

Surface graphs show three-dimensional data where the Z-axis value varies depending on the X- and Y-axis values. Surface graphs are commonly used to graph the results of mathematical formulas.

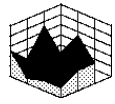
The following graph was created using a mathematical formula:



Option
The following option is available for 3D Surface graphs:

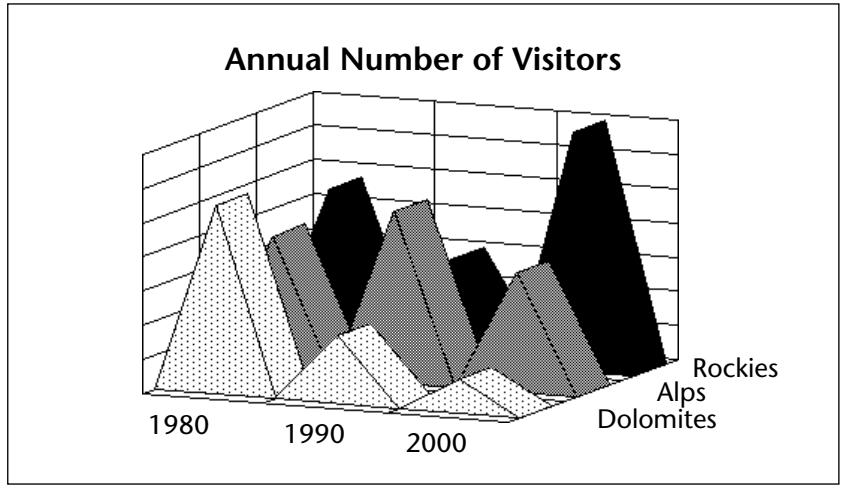
- **Tops only**: Show only the top side of the surface, not the sides. This option is used in the previous example graph.

Here is an example of a surface graph with sides:

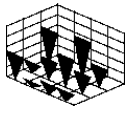


3D Triangle Graphs

3D Triangle graphs compare one item to another, or one or more items over a period of time. 3D Triangle graphs are an alternative to 3D Column graphs.



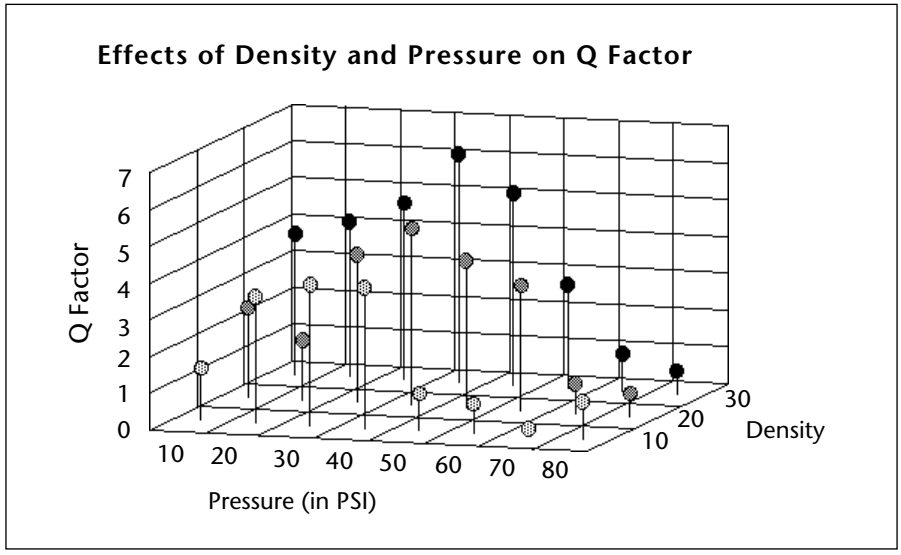
Options
The following options are available for 3D Triangle graphs:



- **Series Gap/Width**: The gap between or width of the series in the graph. The larger the series gap, the smaller the series width.
- **Flipped**: The triangles appear flipped vertically.
- **Plot Zero Values**: If this option is not selected, zero values will be omitted from the graph.

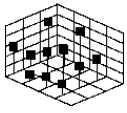
3D Spike Graphs

3D Spike graphs show the intersection of three values. Each data point is connected to the Category plane with a drop line.



Option

The following option is available for 3D Spike graphs:



- **Oval/Square Heads**: Use this option to specify whether the spike heads are oval or square.

Creating a Graph

This section includes the basic steps for creating a graph. After reading this section, you will be able to:

- Create a two-dimensional or three-dimensional graph from data stored in the database or copied to the Clipboard
- Update a graph created from data in the database
- Change a graph's type
- Change the options specific to each graph type.

4D Chart allows you to create two-dimensional and three-dimensional graphs based on the data in fields in your database. You can graph the values in the fields themselves, or you can graph the values that result from formulas that use the fields.

Whatever the case, when you create a graph, you select the data that you want to graph on each axis. For each axis, there are certain restrictions on the data types that can be graphed. If a data type cannot be graphed on a certain axis, fields of that type cannot be selected in the Chart Wizard. The following table provides information on the types of data that can be assigned to the Categories or Series axis and the Values axis.

Data Type	Category or Series Axis?	Values Axis?	Compatible Types on Values Axis
Alpha	Yes	No	-------
Text	Yes	No	-------
Real	Yes	Yes	Integer, Long integer
Integer	Yes	Yes	Real, Long integer
Long integer	Yes	Yes	Real, Integer
Integer 64 bits (*)	No	No	-------
Float(*)	No	No	-------
Date	Yes	Yes	-------
Time	Yes	No	-------
Boolean	Yes	No	-------
Picture	No	No	-------
Blob	No	No	-------

(*) The Integer 64 bits and Float types are only used by the SQL engine of 4D.

Creating a Selection of Records to Graph

Before you begin the process of creating a graph, you must create a selection of records to graph.

4D Chart can operate in its own window or in 4D Chart areas on forms. The following sections explain how to generate the selection of records to be graphed, depending on the 4D Chart location.

Creating a Selection of Records in a Plug-in Window

4D Chart graphs the records in the current selection. Before you create a graph in a plug-in window, select the records you want to graph.

4D Chart will not create a graph unless there is at least one record in the current selection of the table whose data you want to graph.

Creating a Selection of Records in an Input Form

A 4D Chart area in an input form can be used to graph data in other tables.

The table whose records you want to graph must have at least one record in its current selection. If the table that contains the form is automatically related to the table you wish to graph, the current record in the input form determines the selection of records in the related table.

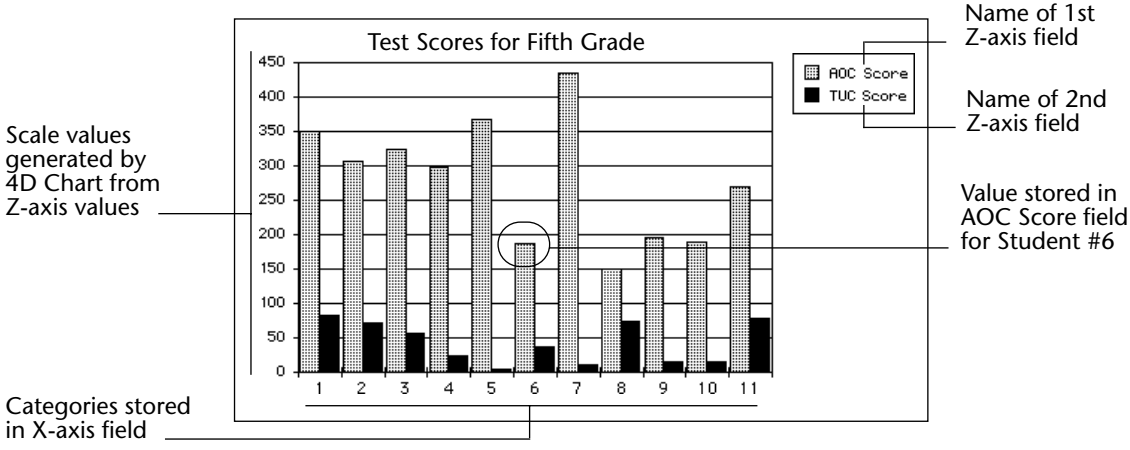
Because you have a record loaded in an input form, you should avoid creating graphs from data in the current table. If you want to graph data from the same table as the input form, you must use PUSH RECORD and POP RECORD or create a new process with the New process function. For more information, see the descriptions of these commands in the *4D Language Reference* manual.

Creating a Graph from Data in the Database

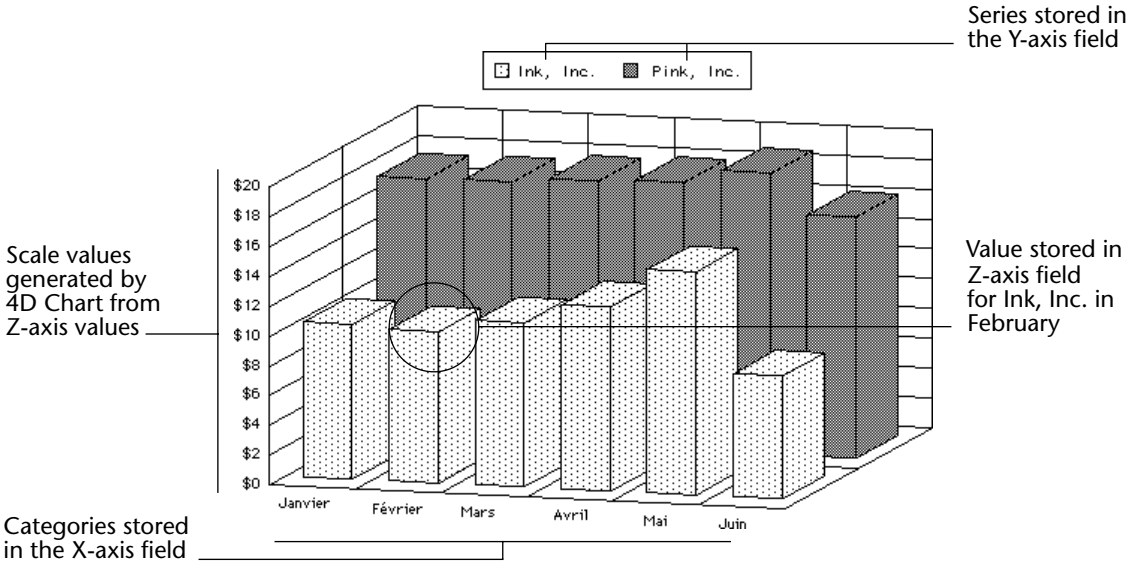
For a two-dimensional graph, you specify a field for the horizontal axis (X-axis) and one or more fields or formulas for the vertical axis (referred to in 4D Chart as the Z-axis).

For a three-dimensional graph, you will specify one field (or formula) each for the X-, Y-, and Z-axes.

The following example graph shows how 4D Chart uses the information in the database to create a 2D graph. The graph plots the scores on two tests for each of 11 students. The X-axis field is Student ID, and the Z-axis fields are AOC Score and TUC Score.



The following example graph shows how 4D Chart uses the information in the database to create a graph. The graph shows the average monthly stock value for two companies. The X-, Y-, and Z-axis fields are, respectively: Month, Company Name, and Average Price.



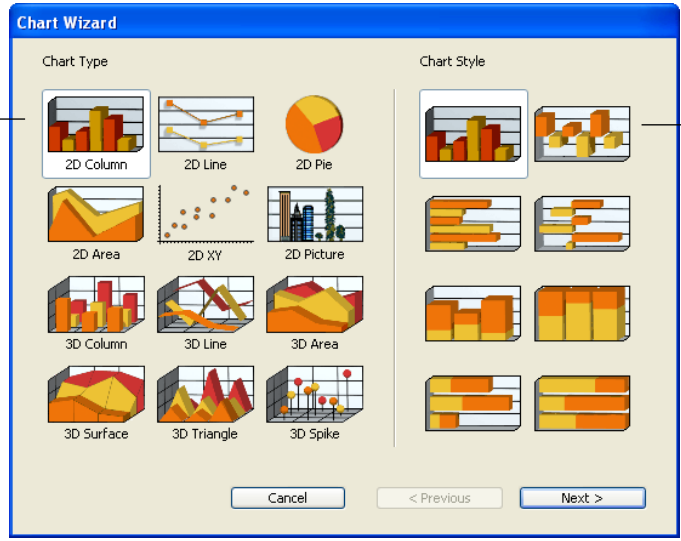
► To create a graph from data stored in the database:

1 **Make sure that no graph is currently selected in the 4D Chart area.**
To deselect a selected graph, click anywhere in the 4D Chart area outside the graph.

2 **Choose the Charts... command in the Tools menu of 4D or click the Chart icon** 

in the 4D toolbar.
OR
If you are in a 4D Chart area, choose New Chart from the Chart menu or click the chart creation icon  **in the 4D Chart tool palette.**

The Chart Wizard appears.



You can choose between 12 types of charts. Types 1-6 are two-dimensional and 7-12 are three-dimensional. By default, the first chart type is selected.

The Chart Wizard has two pages:

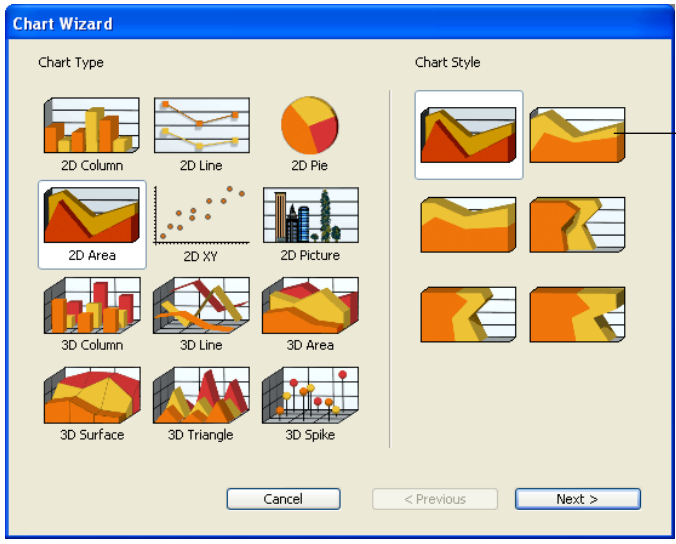
- **Chart Type/Chart Style**: Displays the 12 chart types and the variations on the type of graph that you selected in the Chart Type section. The Chart Style section changes depending on your choice of chart type.
- **Data Selection**: Lets you select the table that contains the data you wish to graph and a field list. Use the Data Selection page to assign fields or formulas to the graph axes. The Data Selection page changes depending on the graph type you chose.

You must choose a chart type and use the Data Selection page to assign appropriate fields or formulas to each axis that the chart type requires.

3 **Click a chart type.**

When you click on a chart type, the Chart Style section and Data Selection page change to match the requirements of the selected type. For complete information on the types of charts available in 4D Chart, see the "Choosing a Graph Type" section on page 1063.

4 Click on a chart style (Optional).



Style variations for 2D area graphs

Note Some chart types allow for only one style.

5 Click the Next > button.

The Data Selection page appears, displaying the appropriate options for the selected chart type. The names of database tables are placed in the "Source Table" drop-down list and the field names of the selected table are displayed in the "Select Fields from the Table" area.

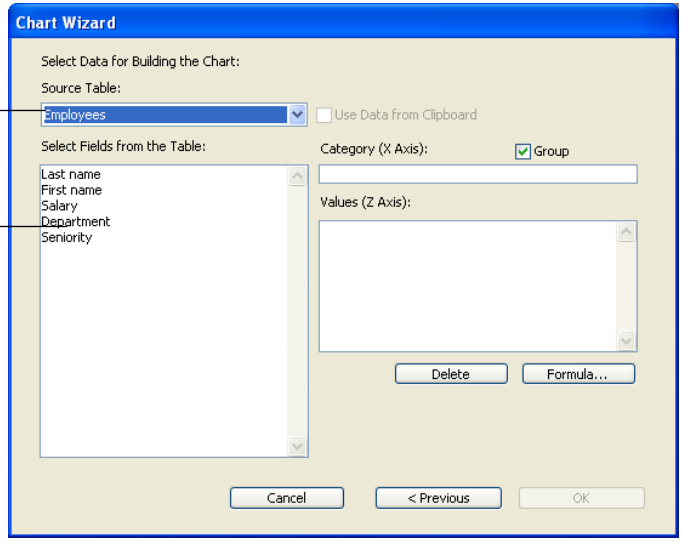


Table drop-down list

Fields list

Note If you are creating a graph in a 4D Chart area in an input form, the current table is not listed in the Tables list because you cannot graph data from the current table.

6 Choose the desired table from the Source Table drop-down list.

The Fields list changes to show the fields from the selected table.

Note Integer 64 bits, Float, Picture and BLOB fields cannot be included in a graphic representation (see the Table "Data Type" on page 1077).

7 Drag the field that you want to assign to the Category axis to the Category box or double-click the field.

Note If you make a mistake when choosing the X-axis field, replace the field by dragging the desired field to the Category box.

- If you want to create a three-dimensional graph, you must fill the "Second Category (Y-axis)" area. Go to the following step.
- If you want to create a two-dimensional graph, you only need to fill the "Values (Z-axis)" area. Go to step 9.

8 If you are creating a 3D graph, drag the Series field to the Y-axis (Second Category) box or double-click the field.

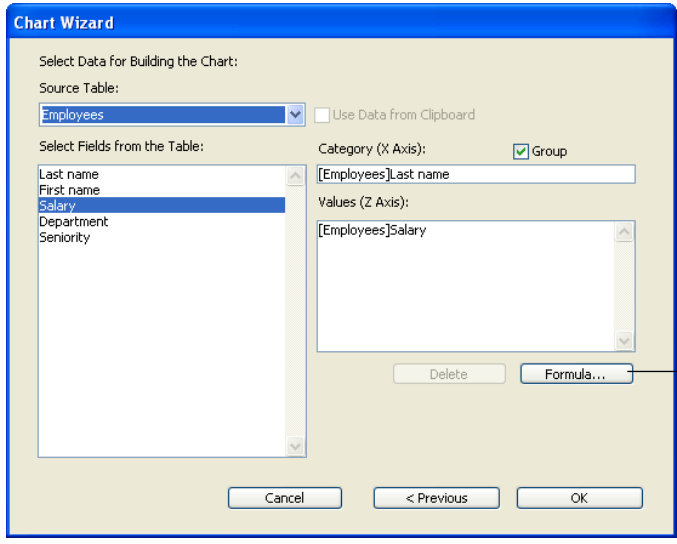
The name of the field appears in the Y-axis box.

Note If you make a mistake when choosing the Y-axis field, replace it by dragging the correct field to the Second Category box.

9 Drag the field containing the values to be graphed to the Values (Z-axis) box or double-click the field[1].
OR
Click the Formula button (2D graphs only).

1. With 2D graphs, you can add more than one field or formula to the Z-axis. Each field or formula becomes a series. For more information, please refer to the "Choosing a Graph Type" section on page 1063.

The name of the field or the formula appears in the Z-axis (Values) box. For instructions on creating a formula, see the "Adding a Formula" section on page 1089.



Button for adding a formula (2D graphs)

For two-dimensional graphs, you can add an unlimited number of fields and/or formulas on the Z-axis (Values). Each item placed in the **Values (Z-axis)** area becomes a series. For more information on series, refer to the "Choosing a Graph Type" paragraph on page 1063.

Notes

• For three-dimensional graphs, to modify the field placed in the Z-axis area, you just need to drag and drop the name of another field onto the area.

• For two-dimensional graphs, you can delete a series, i.e., a field or formula placed in the Z-axis area: just select the series then click on the **Delete** button or press **Backspace**.

10 If you want to have the Z-axis values summed for each X-axis category, click the Group check box.

This option is used when the X-axis categories are not unique and you want each category to appear only once, with the values for each instance summed. For more information about this option, see the "Grouping Non-unique Categories and Series" section on page 1088.

11 **If you want to have the Z-axis values summed for each Y-axis series, click the Group check box (3D graphs only).**

This option is used when the Y-axis series are not unique and you want each series to appear only once, with the values for each instance summed. For more information about this option, see the "Grouping Non-unique Categories and Series" section on page 1088.

12 **When you have finished designing your graph, click OK.**

4D Chart creates the graph and displays it in the 4D Chart area.

Updating the Data in a Graph from the Database

When you create a graph from data stored in the database, the data in the graph is static. Although the data in the database may change, the data in the graph remains the same until you update it.

You can tell 4D Chart to update a graph by choosing **Update** from the **Chart** menu. When you choose Update, 4D Chart regenerates the selected graph using the data in the current selection for the table whose fields you are graphing.

You will want to update a graph in the following situations:

- When you want to include more records in the selection
- When you want to include fewer records in the selection
- When you want to include different records in the selection
- When the data in the selection has been modified.

When you use **Update**, 4D Chart recreates the graph using the new current selection and the settings you chose for the original graph.

Note The **Update** menu item is available only for graphs created from data in the database.

Creating a Graph from Data on the Clipboard

You can graph data from any application if it is correctly formatted and then copied to the Clipboard. This section describes how to format data in order to use it in a graph, and how to create a 4D Chart graph from this information.

Formatting Data and Copying it to the Clipboard

You can graph data that is formatted in the Tab-Tab-Return (TTR) format. If you copy cells from any spreadsheet application, they will be in TTR format. You can also use data from a word-processing application if it is formatted with tabs between the fields and carriage returns between the records.

Following are examples of correctly formatted data:

TTR data (word-processing)

Names → Ages ↵
Helen → 25 ↵
Todd → 27 ↵
Norm → 22 ↵
Michele → 23 ↵

Spreadsheet data

Names	Ages
Helen	25
Todd	27
Norm	22
Michele	23

Note The → and ↵ symbols represent the invisible symbols used by many word-processing applications to indicate tabs and carriage returns, respectively.

The first row of data is used as the field names. Each column contains the data for one field.

Note If you want to use dates copied to the Clipboard, the dates must be in the same format as the System-level date format.

When you have prepared the data in your application, select the desired rows and columns and copy them to the Clipboard.

Creating the Graph from Data on the Clipboard

For a two-dimensional graph, you will specify a field for the X-axis and one or more fields or formulas for the Z-axis. The following illustration shows how 4D Chart interprets the information from the Clipboard.

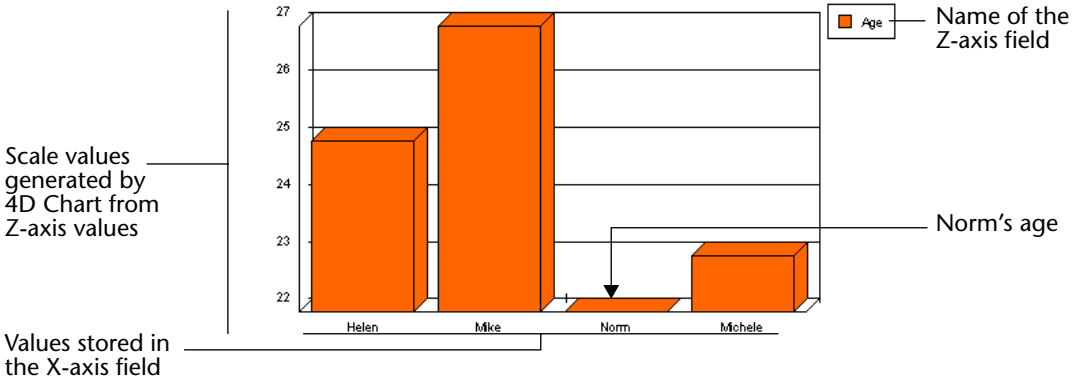
Clipboard

Field names

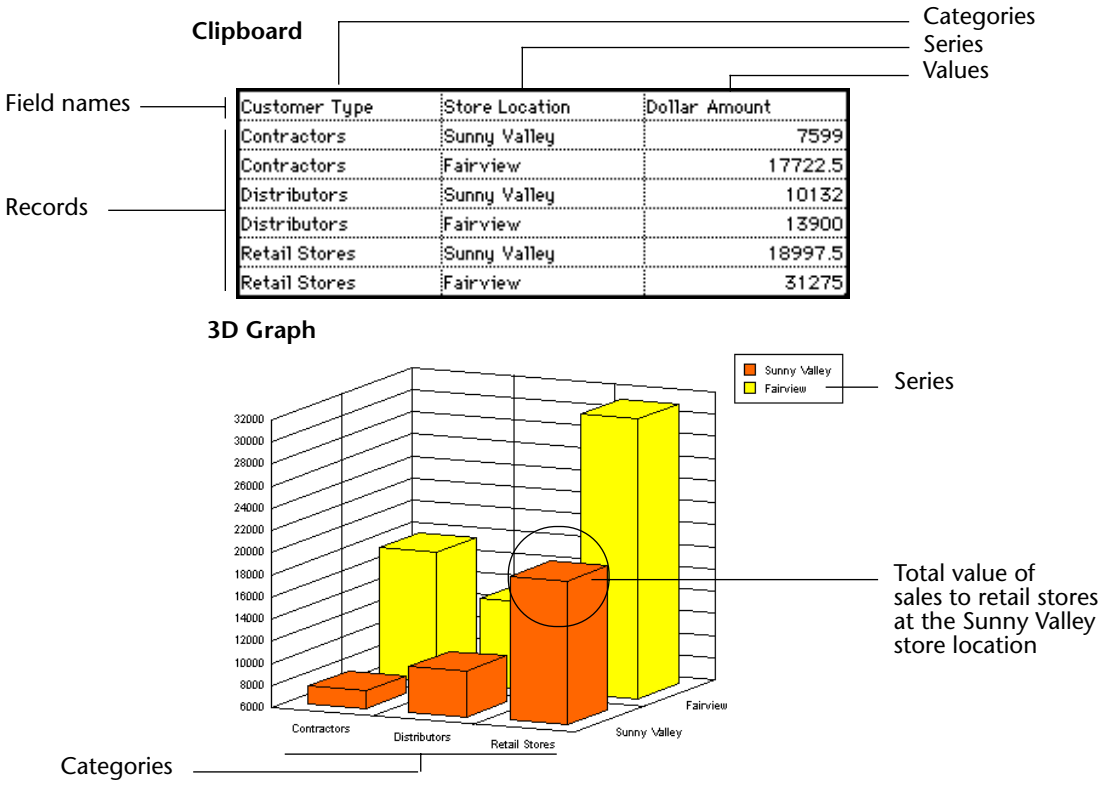
Records

Names	Ages
Helen	25
Mike	27
Norm	22
Michele	23

2D Graph



For a three-dimensional graph, you specify one field each for the X-, Y-, and Z-axes. The following illustration shows the spreadsheet data used for the example graph. The information details the total sales by two different stores to three types of customers.



► To create a graph from data stored on the Clipboard:

1 **Copy the data to be graphed to the Clipboard, using the format described in this section.**

2 **Follow the instructions provided in steps 1 to 5 in the "Creating a Graph from Data in the Database" section on page 1078.**

The Data Selection page appears, configured for the selected chart type.

3 **Click the Use Data From Clipboard check box.**

Clicking Use Data From Clipboard disables the Source Table drop-down list and populates the Fields list with the field names from the Clipboard.

4 **Assign fields to the axes by dragging to the appropriate areas or by double-clicking the fields.**

When all areas have been assigned fields, the **OK** button is enabled.

Note With 2D graphs, you can add as many fields and formulas to the Z-axis as you like. However, be sure to choose only numeric fields. Do not mix date fields and formulas with number fields and formulas. Each item in the Z-axis (Values) box is plotted in the graph.

5 **If you want to have the Z-axis values summed for each X-axis category, click the Group check box.**

This option is used when the X-axis categories are not unique and you want each category to appear only once, with the values for each instance summed. For more information about this option, see the "Grouping Non-unique Categories and Series" section on page 1088.

6 **When you have finished designing your graph, click OK.**

4D Chart creates your graph and displays it in the 4D Chart area.

Updating Graph Data from the Database

When you create a graph using the contents of the database, the data it contains are static. Even when the contents of the database change, the data of the graph can only be modified by an update.

▶ To update a 4D Chart graph:

1 **Choose the Update command in the Chart menu of 4D Chart.**

This way 4D Chart regenerates the selected graph using the data found in the current selection of the table whose fields are being graphed.

You should update a graph in the following cases:

- To add records to the selection,
- To remove records from the selection,
- To put different records in the selection,
- When the data of the selection has been modified.

When you use the update, the graph remains the same with all the previously chosen options, but with a new selection of data.

Note The **Update** menu command is only enabled for graphs created using the contents of the database.

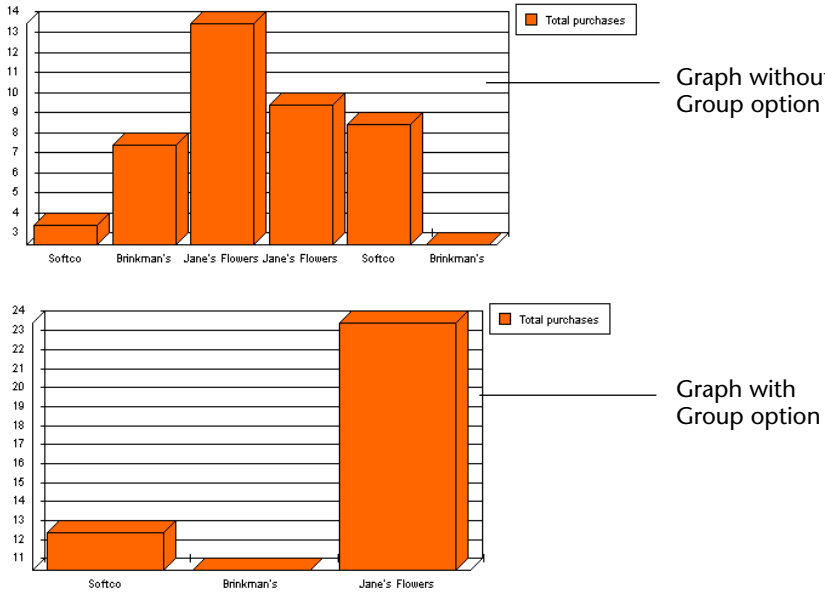
Grouping Non-unique Categories and Series

Some categories or series may appear more than once in your data. For instance, suppose that you want to graph the total purchases made by each of your customers. Each customer may have made several purchases.

When you create a two-dimensional graph, you can choose to sum the values for non-unique categories, so that each category appears only once on the graph. When you create a three-dimensional graph, you can choose to sum the values for non-unique categories, non-unique series, or both.

To create the graph described in the example above, you might graph from the [Invoices] table, using [Invoices]Customer for the X-axis categories and [Invoices]SalesTotal for the Z-axis values. Because some customers may have more than one invoice, you would want 4D Chart to sum the values in the [Invoices]SalesTotal field so that there would be only one value (the total) for each customer.

The illustrations below show the same basic graph with and without the Group option:



To group data on the Category or Series axis, check the corresponding **Group** check box in the Data Selection page of the Chart Wizard.

Adding a Formula

You can graph values that are not represented in your data but are obtained by performing calculations on the data or on other values in your database. For example, you can graph your profits by graphing the results of a formula that subtracts your unit cost from your retail price.

You can refer to any function or valid 4D method "allowed" in the formula editor and graph the result on the Values axis. The method must return a value to 4D Chart. If the categories are grouped, 4D Chart sums the values returned.

If there are fields or other formulas also being graphed on the Values axis, the data types of all the values must be compatible. For instance, if you are graphing a Date field and a formula on the Values axis, the formula must return a date.

Formulas are available only for two-dimensional graphs.

► To use a formula to calculate values for a graph:

1 **When you are ready to specify the Values field in the Data Selection page, click the Formula button.**

The 4D Formula dialog box appears.

2 **Type the formula.**

For more information about the Formula editor dialog box, please refer to "Formula Editor" paragraph on page 981. Following are some examples of formulas:

Formula	Syntax
Month of (Current date)	4D function (4D function)
Day of ([Invoices]Invoice date)	4D function (field)
Sin (vX)	4D function (variable)
NumVisits	Method that returns a value in $0 This method must have been "allowed" in the Formula editor using the SET ALLOWED METHODS command

3 **Click OK.**

4 **When you have finished designing the graph, click OK in the Chart Wizard to create the graph.**

Changing the Chart Type

You can switch between chart types at any time. You can change from one two-dimensional graph type to another, or from one three-dimensional graph type to another. When creating a graph, feel free to experiment with chart types to find the best way to present your data.

► To change the chart type:

1 **Make sure that the graph is selected.**

You select a graph by clicking it. When a graph is selected, selection handles appear around it.

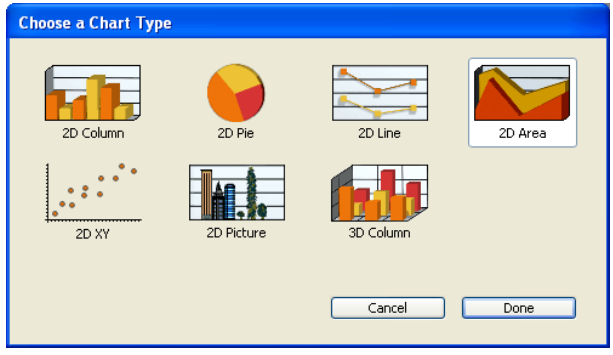
2 **Click the Chart button in the 4D Chart Tool palette.**

A pop-up menu appears, which lets you select another type of chart. This menu does not allow a preview of the chart types.

OR

Choose the Chart Type item in the Chart menu of 4D Chart.

The "Choose a Chart Type" window is then displayed.



You can select another chart type and click **OK**.

The graph is redrawn using the new chart type. Both the Chart drop-down list and the Choose a Chart Type window display only the chart types that are appropriate for the selection of data. You cannot, for example, plot 3D data using a 2D chart type. For information on the graph types available in 4D Chart, see the "Choosing a Graph Type" section on page 1063.

Setting the Chart Type Change Alert

You can have 4D Chart display an alert dialog box when the user attempts to change the type of a graph. The user then has the option to cancel or continue with the change.

► To display an alert dialog box when a new chart type is selected:

1 **Choose Properties from the 4D Chart Edit menu.**
The Properties dialog box appears.

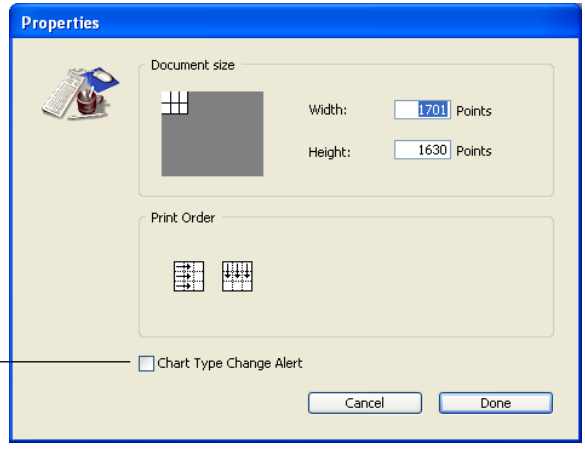


Chart Type Change Alert check box

2 **Check or uncheck the Chart Type Change Alert check box.**

3 **Click OK to close the Properties dialog box.**

Changing the Options for a Chart Type

Each chart type has a particular set of options that you can change using the Options dialog box. By changing the chart options, you can, for example, change a column graph to a bar graph, or show the series in an area graph as proportions of a whole.

► To open the Options dialog box for a graph:

■ **Double-click the graph.**
OR
Select the graph and choose Options from the Chart menu.

For more information about the options for each chart type, see the "Choosing a Graph Type" section on page 1063.

Modifying Graph Features

This section explains how to modify graph features. After reading this section, you will know how to:

- Resize a graph
- Customize graph axes
- Show and hide grid lines
- Display the series values
- Customize legends
- Add depth to a two-dimensional graph
- Change the perspective of a three-dimensional graph
- Customize the tips of a graph
- Change the graphic attributes of graph objects
- "Explode" a wedge from a pie chart
- Add a picture to a picture chart.

Resizing a Graph

When you create a graph, its initial size corresponds to the size of the window in which it is created.
If the graph is created using the **Tools>Chart...** menu command of 4D, its size corresponds to the default size of a 4D Chart external window . If the graph is created from a 4D Chart window, its size corresponds to that of the window.

You can resize a graph in two ways:

- By resizing the window in which it is displayed.
A graph is "unified" with the window where it was created as long as its own dimensions have not been modified using the selection handles (see below).
- By directly resizing the graph in the window using the selection handles (black squares that appear around the object when it is selected). Once you have done so, the graph is no longer "unified" with the window (see above).

You can resize the height and width at the same time, maintaining the sizing proportion between the two.

► To resize a graph:

1 **Select the graph.**

2 **Hold down the mouse button on a selection handle and drag it up, down, or diagonally.**

If you drag a corner, the height and width change. If you drag a side, either the height or the width changes.

To resize the object so that the height and width retain the same proportion to each other, hold down the **Shift** key while you are resizing.

3 **Release the mouse button when you have finished.**

The graph is resized.

Customizing the Axes

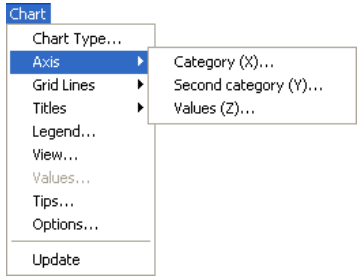
You can customize many aspects of each axis of a graph. This section contains information about:

- Customizing the axis labels
- Customizing the tick marks
- Changing the scale of the Values axis
- Reversing the order of data points on an axis
- Positioning the origin
- Adding axis titles.z

The following table explains the axes available in two-dimensional (2D) and three-dimensional (3D) graphs:

Number of Axes	Axis Name	Axis
Two (2D Graph)	Category	X
	Series	N/A
	Values	Z
Three (3D Graph)	Category	X
	Series	Y
	Values	Z

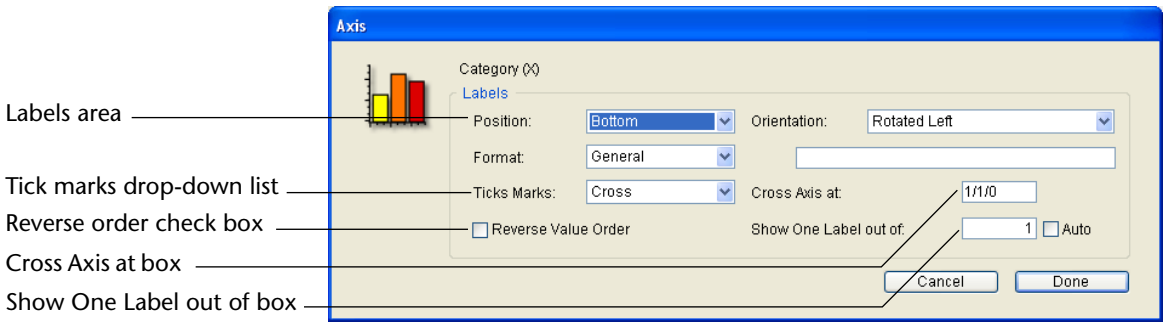
You can display the Axis dialog box for any axis by choosing the appropriate axis from the **Axis** submenu of the **Chart** menu.



Using the Axis dialog box, you can change the location of the axis labels, tick marks, and origin. You can also change the scale used for the Values axis and reverse the order of the items graphed on each axis.

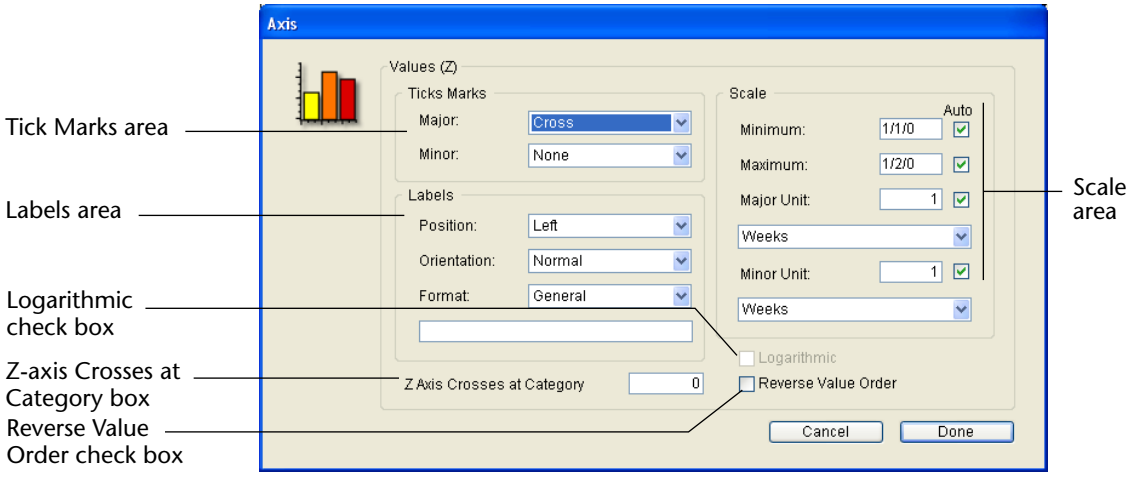
This section presents the dialog boxes that you use to customize the axes of your graph. It then provides details about each of the options.

The Category Axis dialog box enables you to customize the Category and Series axes:



Note The Tick Marks drop-down list, Reverse Order check box, and Cross Axis At text box are not available in the Category or Series Axis dialog box for three-dimensional graphs.

The following dialog box enables you to customize the Values axis:



Note Only the Labels and Scale areas are available in the Axis dialog box for three-dimensional graphs. The Date Increment drop-down lists are available only when dates are being graphed on the Values axis.

Customizing the Axis Labels

4D Chart automatically labels the axes when it generates a graph. You can change the position, orientation, and format of the labels, or you can decide not to display them with your graph.

- **Label Positions**
  You can choose one of the following label positions from the **Position** drop-down list:
  - None
  - Top
  - Bottom
  - Left
  - Right.

  Depending on the orientation of the axis, either Top and Bottom or Left and Right will be unavailable.

  When the position is None, the label does not appear on the graph.

- **Label Orientation**
  The following label orientations are available for each axis:

	Orientation					
	Normal	**Vertical**	**Rotated Left**	**Rotated Right**	**Staggered**	**Wrap**
Result	Label	L a b e l	Label	Label	Label1 Label3 Label2	Lab el

- **Label Formats**
  You can change the way data in your labels is displayed by using display formats. For instance, you can use a display format to display dollar amounts using the dollar sign ($), commas, and decimals.

  The following table shows some examples of the effects of display formats:

Data in Default Format	Display Format	Data in Display Format
3400	$###,###.00	$3,400.00
3/4/07	Month Date, Year	March 4, 2007

  When you select one of the formats from the Format drop-down list, it is entered into the Format text box, located to the right of the drop-down list. If you want to use a custom format, you can enter it in that area.

  You can use one of 4D's built-in formats, edit a format, or create one of your own. For more information, see the "Display Formats" paragraph on page 521.

- **Labeling Data from Boolean Fields**
  If you create graphs using Boolean fields from the database, the default axis labels are "0" and "1", corresponding to FALSE and TRUE. You can label your graph more informatively by changing the label format.

  To create more meaningful labels for Boolean fields, change the label format to the following:

  TrueLabel;;*FalseLabel*

  For example, you might change the format to "Female;;Male" or "Experimental;;Control".

Reducing the Number of Labels

You can reduce the number of labels displayed on the X-axis or Y-axis of a graph. This feature is useful when you create graphs with a large number of categories (more than 100).

When this option is used, 4D Chart displays only one label per N labels on the selected axis. To use this option, enter a value between 2 and 255 in the "Show One Label out of" box. For example, if you enter 10, 4D Chart will display one label per 10 categories. The value 1 is the standard setting (all labels are displayed).

- **Auto**: If you check the **Auto** box, 4D Chart will calculate the appropriate number of labels to display, according to the amount of room.

Customizing the Tick Marks

Tick marks show the increments of the axes. You can choose among different styles of tick marks, or you can decide not to show the tick marks. You can customize this option for each axis in a two-dimensional graph.

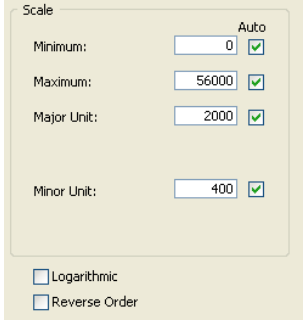
Note You cannot customize the tick marks for a three-dimensional graph.

The following tick mark styles are available:

- Cross (⊣⊢)
- Inside (⊣—)
- Outside (—⊢).

Changing the Values Axis Scale

You can change the minimum and maximum values. By default, 4D Chart uses the minimum and maximum value of the set of data being graphed. You can change these values, for example, to start the Z-axis at zero.



You can also modify the major and minor units for tick marks. By default, 4D Chart calculates these parameters based on the number of values to be shown and the size of the graph. If the values are dates, you can specify whether the major and minor units are measured in days, weeks, months or years.

For each scale display option, there is an **Auto** option which lets you enable the default operation of 4D Chart.
If you want to use custom values, remember to uncheck the **Auto** option, otherwise your parameters will not be taken into account.

Note The major and minor unit grid lines can be either hidden or displayed (see the "Showing and Hiding Grid Lines" paragraph on page 1100).

- **Using a Logarithmic Scale**
  You can change the scale from normal to logarithmic by checking the **Logarithmic** check box, located in the Values axis dialog box.

Reversing the Order of the Data

You can reverse the order of the data on an axis by checking the **Reverse Order** check box. This can only be done in a two-dimensional graph.

Positioning the Origin

You can change the position of the origin — the place where one axis crosses another — for each axis in a two-dimensional graph.

- **Changing the Origin of the Values (Y) Axis**
  The origin of the Z-axis is the value at which the Category (X) axis crosses the Z-axis. Usually, the origin is the minimum value on the graph or zero. Sometimes you will want to change the position of the origin.

▶ To change the origin of the Values axis:

1. **Choose Category (X) from the Axis submenu of the Chart menu.**
   The Category (X) dialog box appears.
2. **Enter a value in the "Cross Axis At" area.**
   The value you enter is the numerical value of the desired origin position.
3. **Click OK to close the dialog box.**

- **Changing the Origin of the Category (X) Axis**
  The origin of the X-axis is the category at which the Values axis crosses the X-axis. Usually, the origin is placed to the left of the first category. Sometimes you will want to move the origin.

► To change the origin of the Category (X) axis:

1 **Choose Values (Z) from the Axis submenu of the Chart menu.**
The Values (Z) dialog box appears.

2 **Enter a number in the "Z Axis Crosses at Category" area.**
The number you enter is the number of the category at which the origin should cross. The categories are numbered from left to right, (or from bottom to top in a horizontal graph). The Z-axis crosses to the left of the specified category.

To move the origin to the right of the last category on the graph, specify the number of categories plus 1.

If the specified number is higher than the number of categories plus 1, the value is ignored and the origin is reset to the left of the first category.

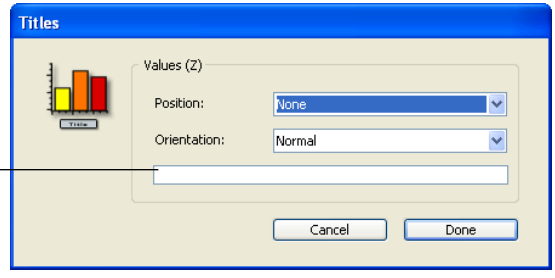
3 **Click OK to close the dialog box.**

Adding Axis Titles

You can add a title for each axis. Usually, the titles describe the data graphed on the axis.

► To add a title to any axis:

1 **Choose the appropriate axis from the Titles submenu of the Chart menu.**
The Titles dialog box appears.



Title text box

2 **Type the title into the "Title" text box.**

3 **Choose a position for the title from the "Position" drop-down list.**
This specifies the position of the title relative to the graph.

You can choose one of the following title positions:

- None
- Top
- Bottom
- Left
- Right.

The default position is None. When the position is None, the title does not appear on the graph.

Depending on the orientation of the axis, either Top and Bottom or Left and Right will be unavailable.

4 **Choose an orientation for the title from the "Orientation" drop-down list.**

The following orientations are available for each axis:

- Normal
- Vertical
- Rotated Left
- Rotated Right.

5 **Click OK.**

Showing and Hiding Grid Lines

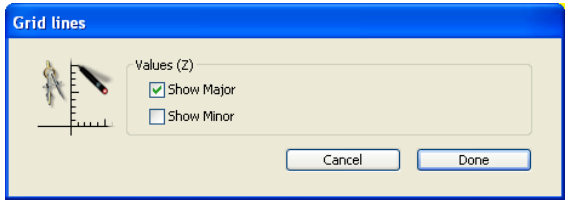
You can display grid lines for the major and minor increments of each axis. Grid lines can make a graph easier to read.

The major and minor increments are set in the Axis dialog box for the Values axis.

► To display the grid lines for any axis:

1 **Choose the appropriate axis from the Grid Lines submenu of the Chart menu**

The Grid Lines dialog box appears.



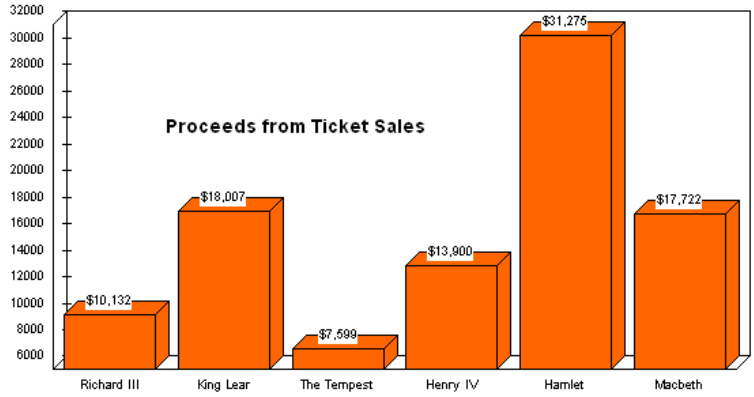
2 **Click the desired check boxes and click OK.**

Displaying the Series Values

You can display information for each series directly in the graph. You can display the following information:

- **Values**: The actual number or date, in Arabic numerals. You can show the values on the graph instead of using labels on the Values axis, or use them to supplement the axis labels.
- **Percentages**: The value of the data point divided by the sum of all the values in the category, as a percentage.
- **Categories**: The name of the category, identical to the axis label for the category.

The following graph shows the values at the tops of the columns:



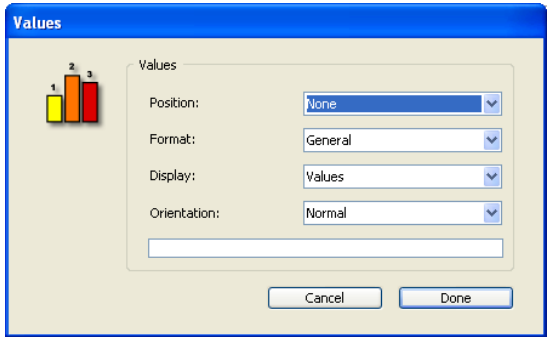
You can select the location at which the values, percentages, or category labels should appear, as well as their orientation and format.

This feature is not available for three-dimensional graphs.

▶ To display information about the data in a series:

1 Choose Values from the Chart menu.

The Values dialog box appears.



2 Select the location at which the information should appear.

You can select one of the following options from the **Position** drop-down list:

Pie Charts	All Other Graph Types
None	None
Inside	Outside Top
Outside	Outside Bottom
	Inside Top
	Inside Centered
	Inside Bottom
	At Axis

3 Select the type of information you wish to display.

You can select one of the following options from the Display drop-down list:

- Values
- Percentage
- Category
- Value & Percent
- Categories & Percent.

4 If you wish, select a format from the "Format" drop-down list.

You can change the way values are displayed by using display formats. For instance, you can use a display format to display dollar amounts using the dollar sign ($), commas, and decimals.

The following table shows some examples of the effects of display formats:

Data in Default Format	Display Format	Data in Display Format
3400	$###,##.00	$3,400.00
3/4/07	Month Date, Year	March 4, 2007

When you select one of the formats from the Format drop-down list, it is entered into the Format text box, located below the drop-down list.

You can use one of 4D's built-in formats, edit a format, or create one of your own. For more information, see the "Display Formats" paragraph on page 521.

5 **Select an orientation from the "Orientation" drop-down list.**
You can select one of the following options:

- Normal
- Vertical
- Rotated Left
- Rotated Right.

6 **When you are finished making your selections, click OK.**

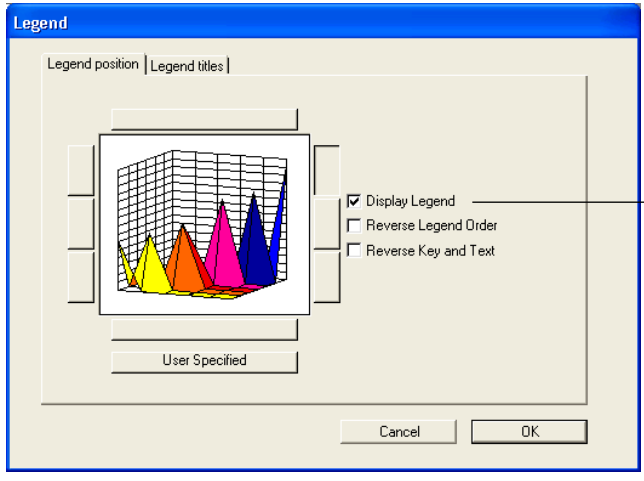
Customizing the Legend

4D Chart automatically creates a legend for you when it generates the graph. By default, the legend is based on value labels. You can customize the legend's location, order, and text.

Displaying and Hiding the Legend

To display or hide the legend for the selected graph:

1 **Choose Legend from the Chart menu.**
The Legend dialog box appears.



2 **Click the Display Legend check box.**
If this check box is checked, the legend is displayed as part of the graph object. If this check box is unchecked, the legend is hidden.

3 **Click OK to close the dialog box.**

The next section explains how to position the legend in the graph object.

Positioning the Legend

The legend is part of the graph object. You can position the legend using the eight built-in legend locations, or you can move it using the mouse. You can move the legend anywhere in the document, including placing it inside the graph itself.

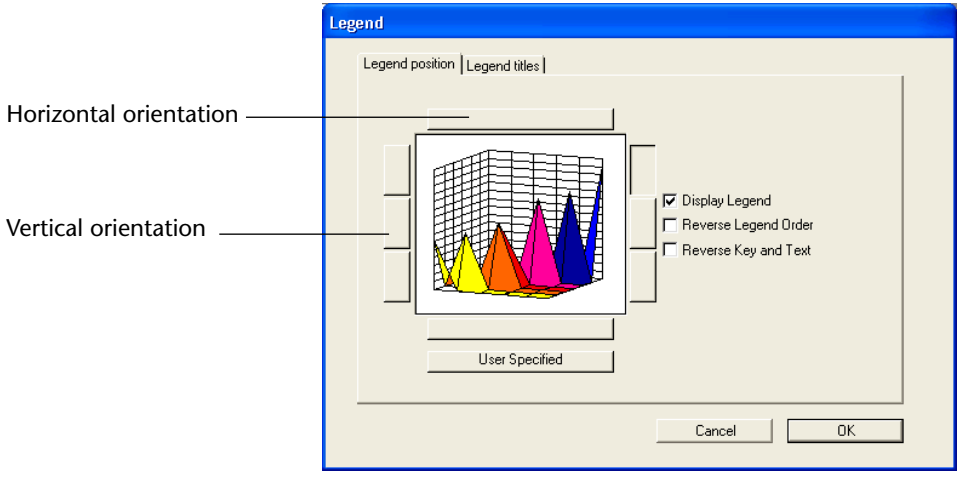
In addition, you can display the legend vertically or horizontally. In other words, the series in the legend can be placed left-to-right or top-to-bottom.

Positioning the Legend Using the Built-in Locations

To use the built-in locations for the legend:

1 **Choose Legend from the Chart menu.**

The Legend dialog box appears.



2 **Select a legend position by clicking one of the model legends in the Location area.**

The location you choose will determines whether the orientation is horizontal or vertical.

3 **Click OK to close the dialog box.**

Positioning the Legend with the Mouse

You can position the legend anywhere in the 4D Chart document by moving it with the mouse.

► To position the legend with the mouse:

1 **Hold down the Ctrl key (Command key on Macintosh) and click the legend to select it.**

2 **While continuing to hold down the Ctrl or Command key, click and drag the legend to the desired location.**

3 **Release the mouse button and the Ctrl or Command key.**

Maintaining the Legend Location

When you use the Legend dialog box, click the **User Specified** button to maintain the legend location. If you select one of the built-in locations, the **User Specified** option is automatically deselected.

Setting the Legend Order

You can reverse the order of the series in the legend. In addition, you can reverse the order of the legend key and legend text.

► To change these options:

1 **Choose Legend from the Chart menu.**

The Legend dialog box appears. The following options are available:

- **Reverse Legend Order**: Reverses the order of the series in the legend.
- **Reverse Key and Text**: If this check box is checked, the square containing the color or pattern code for each series is displayed after the series name.

2 **Check or uncheck the check boxes as desired.**

3 **Click OK to close the dialog box.**

Customizing the Legend Text

You can customize the text of any or all of the series in a legend.

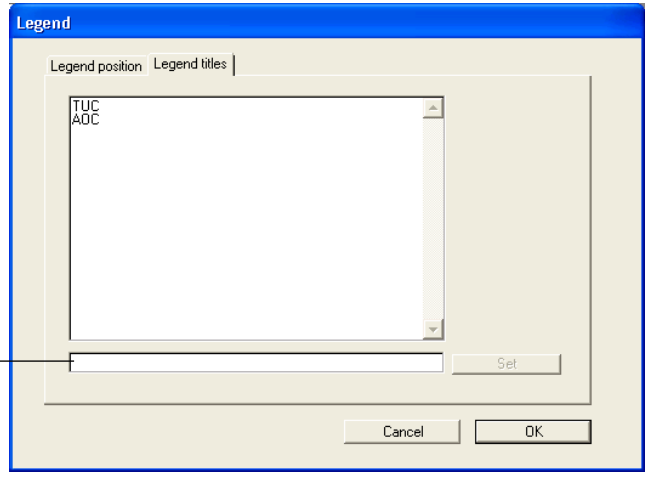
► To customize legend text:

1 **Choose Legend from the Chart menu.**

The Legend dialog box appears.

2 **Click the Legend Titles tab.**

The Legend Titles page appears.



The series labels for the legend are displayed in a list.

3 **Choose the series label to edit by clicking its name in the list.**

The series label is highlighted in the list and the text of the label appears in the Text Editing box.

4 **Edit the text in the Text Editing box.**

5 **Click the Set button.**

You must click **Set** in order for the changes to take effect.

6 **Repeat the previous steps until you have made all the changes you want.**

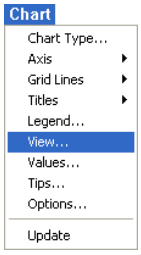
7 **Click OK.**

The Edit Legend dialog box closes. If you are displaying the legend, you will see your changes in the legend text.

Modifying Depth in a Two-dimensional Graph

A three-dimensional graph plots three fields or variables in three dimensions. However, you can add the appearance of a third dimension to a two-dimensional graph by adding "depth." The third dimension does not represent the values of any of the fields or formulas plotted in the graph.

By default, 4D Chart adds the illusion of depth to two-dimensional graphs. You can remove it to make the values in the graph easier to read, or modify the three-dimensional illusion.

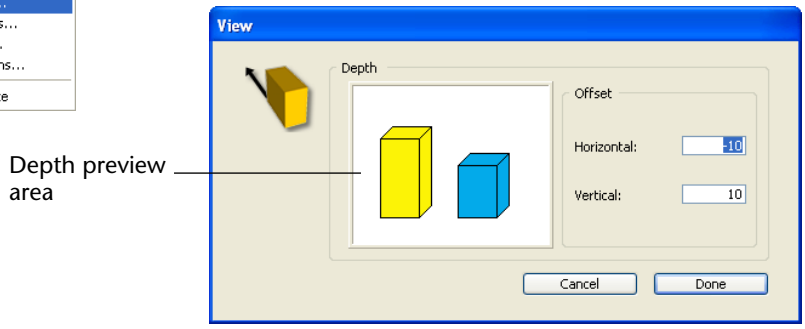


► To modify or remove the depth illusion in a two-dimensional graph:

1 **Choose View from the Chart menu.**

This menu item is enabled only when the currently selected object is a graph.

The Depth dialog box appears.



Depth preview area

2 **Modify the horizontal and vertical depth, measured in points, by typing values in the appropriate text boxes.**

To remove the depth illusion, enter zeros in both entry areas. You can also enter negative values to invert the axis of the three-dimensional illusion.

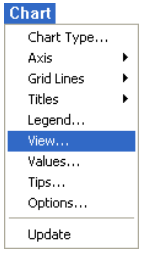
When you press **Tab** or click outside a text box, the value you have entered is reflected in the Depth preview area.

3 **Click OK.**

Your Depth specifications are applied to the graph.

Changing the Perspective of a Three-dimensional Graph

You can change the perspective from which a three-dimensional graph is viewed. You can change both the rotation and the elevation of a graph.

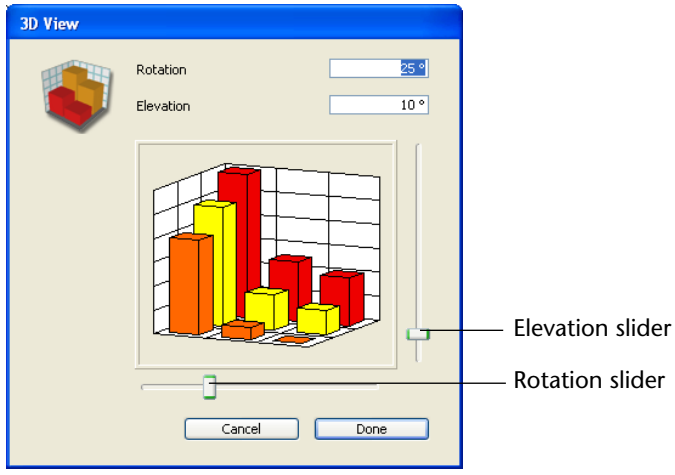


► To change the perspective from which a three-dimensional graph is viewed:

1 **Choose View from the Chart menu.**

The View menu item is enabled only when the currently selected object is a graph.

The 3D View dialog box appears.



2 **If you wish, you can change the rotation by entering a new value (from 0 to 90) in the Rotation text box or by moving the Rotation slider to the left or right.**

Rotation is the rotation of the graph around the Values axis.

The model graph reflects the changes you have made.

3 **If you wish, you can change the elevation by entering a new value (from 0 to 90) in the Elevation text box or by moving the Elevation slider up or down.**

Elevation is the rotation of the graph around a horizontal line perpendicular to the Values axis.

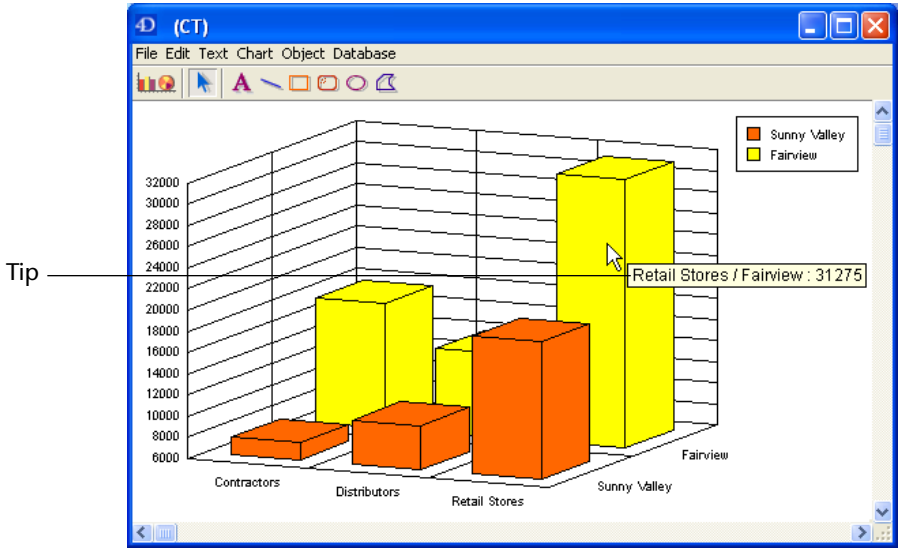
The model graph reflects the change you have made.

4 **Click OK.**

Your graph is redrawn from the new perspective.

Customize Tips in Graphs

Tips — information about a graph and its data — are available for both XY and non-XY graphs. The user can display tips from any position of the pointer in the graph.



Tips display the following types of information:

- Value (on which the pointer is located).
- Ratio between a value (on which the pointer is located) and the total of values in that category. This ratio is expressed as a percentage.
- Category.

The following table lists the graph types and the information that can be displayed in the associated tips:

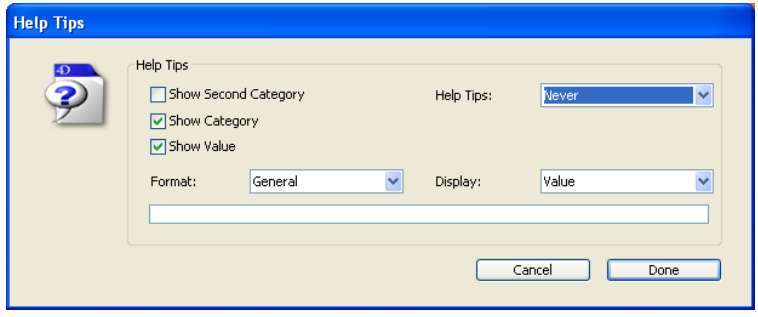
Type of Graph	Tips
2D Column	Values and percentages
2D Line	None
2D Pie	Values and percentages
2D Area	None
2D XY	Values only
2D Picture	Values and percentages
2D Polar	Values only
3D Column	Values only
3D Line	None
3D Area	None

Type of Graph	Tips
2D Column	Values and percentages
3D Surface	None
3D Triangle	Values only
3D Spike	Values only

The values in the tips are based on the data as displayed in the graph and thus may be approximations based on the screen resolution.

Tip attributes are accessed manually in the Tips dialog box (from the **Tips** item in the **Chart** menu) or programmatically using the CT GET TIPS ATTRIBUTES and CT SET TIPS ATTRIBUTES commands.

The Tips dialog box appears as follows:



Note In the case of XY type graphs, the dialog box contains a second **Format** drop-down list and an additional entry area for the Y-axis.

In the Tips dialog box, you can change the properties of the tips displayed for any graph.

The Tips list box enables you to select when to display tips. The choices are **Never**, **Always,** and **On Request**.

- If you select **Always**, tips appear when the user places the mouse cursor over part of the graph.
- If you select **Never**, tips are disabled.
- If you select **On Request**, tips will appear when the user places the mouse cursor over part of the graph while holding down the **Ctrl** (Windows) or **Command** (Mac OS) key.

By default, the **Never** option is selected.

Selecting the **Show Second Category** displays the value of the second Category axis in the tips. The default is to not show the second category. In the case of a 2D graph, selecting this option displays the field name of the Category axis value.

Selecting the **Show Category** option displays the value of the Category axis in the tips. The default is to display the Category axis value.

Selecting the **Show Value** option displays the value of the Values axis in the tips. The default is to display the Values axis value.

The **Format** list box enables you to select the display format for values displayed. The default is General. If the Format selected is General, the text box below the Format list box allows you to enter text for the Values axis. In the case of XY graphs, there are two Format list boxes and text boxes, to accommodate the X- and Y-Values axes.

The **Display** list box enables you to choose to display the category value, percentage (the ratio between the value on which the pointer is located and the total of values in that category), or both.

Changing the Attributes of Graph Objects

This section describes how to select individual graph objects (axis lines, grid lines, series elements, etc.) and change their graphic attributes, such as color, pattern, and line width. You can also change the font attributes of graph text objects (such as axis labels and titles).

► To select an individual graph object:

- **Hold down the Ctrl key (Windows) or Command key (Macintosh) and click the object.**

  The following graph objects can be selected:

  - Each series in a two-dimensional graph
  - Each of the three visible sides of a series in a 3D graph

Note To select all sides of a series in a three-dimensional graph at once, hold down the **Shift** and **Ctrl** keys (Windows) or **Command** key (Macintosh) and click an object in the series.

  - Each axis (includes the tick marks)
  - Axis labels for each axis
  - Major grid lines for each axis
  - Minor grid lines for each axis

- Titles for each axis
- The legend
- Values displayed for the series.

Changing Object Attributes

After you have selected an object, you can modify its graphic attributes using items in the **Object** menu.

Each of the following object attributes can be changed in the **Object** menu.

- **Fill Pattern:** The pattern displayed inside the border of an object. All objects except lines have fill patterns.
- **Fill Color**: The color displayed inside the border of an object.
- **Line Pattern**: The pattern of a line or border. The default line pattern is solid.
- **Line Color**: The color of a line or border. The default line color is black. The color palette used for Fill Color is also used for Line Color.
- **Line Width**: The width of a line or border measured in points. The default line width is 0.25 points (also called a hairline).

Changing Text Attributes

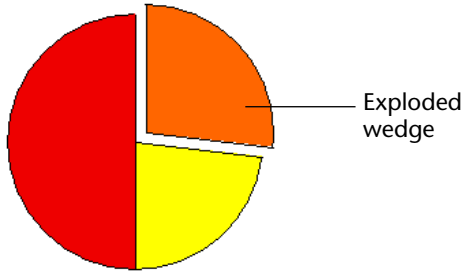
Text attributes apply only to axis labels, axis titles, legend text, and series values. Each of the following text attributes can be changed in the **Text** menu:

- **Font**: The typeface of the text.
- **Size**: Font size is measured in points.
- **Style**: Styles include plain, bold, and italic. The default style is plain.
- **Color**: The default color is black.

You cannot change the justification of graph text. You can only change the justification of text added with the Text tool. For more information about adding text with the Text tool, see the "Adding Text" section on page 1118.

Exploding Wedges from a Pie Chart

You can “explode” a pie chart by pulling one or more wedges away from the center of the pie.



► To explode a wedge from a pie chart:

1 **Hold down the Ctrl key (Windows) or Command key (Macintosh) and click a wedge of the pie chart to select it.**

Selection handles appear around the wedge.

2 **While holding down the Ctrl key (Command key on Macintosh), click and drag the wedge away from the center of the pie.**

3 **When the wedge is where you want it, release the mouse button and the Ctrl key (Command key on Macintosh).**

Adding Pictures to a Picture Chart

When you create a picture chart, the columns are filled with a default picture. You can add your own picture for each series by pasting it from the Clipboard.

► To paste a picture into the columns for a series:

1 **Make sure that the picture you want to paste into the column is on the Clipboard.**

2 **Hold down the Ctrl key (Windows) or Command key (Macintosh) and click a column of the desired series.**

All the columns are then selected. By default, the picture for each series shows a building.

3 **Choose Paste from the 4D or 4D Chart Edit menu.**
OR
Press Ctrl+V under Windows or Command+V under Mac OS.

The picture is pasted into each column for the series.

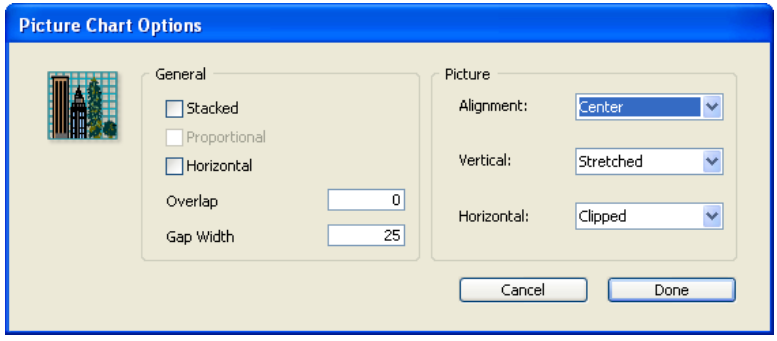
You can repeat this process for each series.

Note To recover the standard picture, apply a different chart type then re-apply the 2D Picture type.

Adjusting the Pictures within the Columns of Picture Charts

To customize the proportions of the pictures within the series columns:

1 **Double-click the graph to display the Options dialog box for picture graphs.**
OR
Select the graph and choose Options from the Chart menu.
The Picture Chart Options dialog box appears.



In addition to the chart options, which are discussed in the "Choosing a Graph Type" section on page 1063, you will find the following options for aligning and adjusting the pictures:

- **Alignment**: This option specifies the horizontal alignment of the picture: center, left, or right.
- **Vertical**: This option specifies how the picture uses the vertical space of the column. The options are: clipped, stretched, and stacked. For more information about these options, see the table at the end of this section.
- **Horizontal**: This option specifies how the picture uses the horizontal space of the column. The options are: clipped, stretched, and stacked. For more information about these options, see the table at the end of this section.

The following table explains the Vertical and Horizontal options.

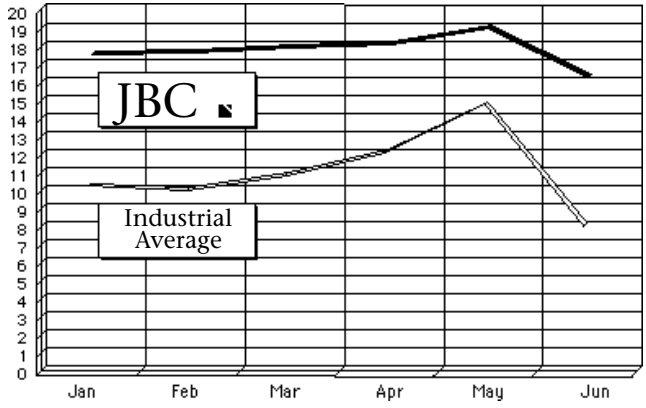
Option	Description
Clipped	If the picture is too tall or too wide to be displayed in its entirety, it is truncated at the edges of the column. The proportions of the picture remain unchanged.
Stretched	The picture is stretched or shrunk so that its dimensions match those of the column.
Stacked	If the picture is either too short or too narrow to fill the column, the picture is repeated until the column's height or width is reached. When the edge of the column is reached, the picture is truncated.

2 Choose the desired options from the drop-down lists and click OK.

Adding Objects and Text

With 4D Chart you can add a variety of static objects to your documents (lines, rectangles, ovals, polygons) as well as text.

The following graph uses objects and text to identify the series.



You can draw the following objects:

- Lines
- Rectangles
- Rounded rectangles
- Ovals
- Polygons .

Selecting a Drawing Tool To draw an object, you must first select a tool from the Object Tool palette. You make a tool active for drawing when you select it.

The mouse pointer changes depending upon its use. If the Arrow tool is selected, then the pointer is an arrow ↖. You use the Arrow tool to select menu items and objects.

When you select any of the graphic object tools, the pointer changes to a crosshair +. You use the crosshair to draw graphic objects.

Locking a Drawing Tool When you select a tool, it is in effect only while you draw one object, after which the Arrow tool is selected. However, you can *lock* a tool so that you can continue to use it as long as you need it, by double-clicking it. When you lock a tool, it remains in use until you select another tool.

Drawing an Object

► To draw all objects except polygons:

1 **Select a tool in the 4D Chart toolbar.**

1 **Hold down the mouse button in the document area and drag the mouse to draw the object.**

2 **Release the mouse button to finish drawing the object.**

► To draw a polygon:

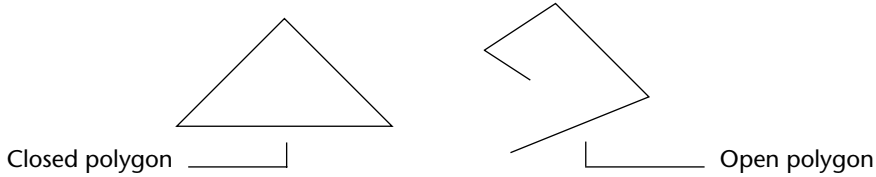
1 **Click to anchor the first vertex.**

2 **Drag the mouse to draw a side and then click to anchor the next vertex.**

3 **Continue anchoring vertices and drawing sides as necessary.**

4 **Close the polygon by clicking the first vertex to close the polygon, or by pressing Alt+Ctrl+Enter (Windows) or Option+Command+Enter (Mac OS).**
OR

Leave an open side to the polygon by double-clicking the mouse button to anchor the last vertex or by pressing Enter.



Constraining Graphic Objects As You Draw

By constraining graphic objects as you draw, you can control the drawing process — in particular, the height and width of objects. For example, a square is really just a constrained rectangle.

The following table lists the effects of constraining each object:

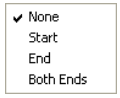
Object	Key	Constraint
Lines	Shift	Constrain line to 45° angles
Rectangles	Shift	Draw a square
	V (for *vertical*)	Constrain height
	H (for *horizontal*)	Constrain width
Rounded Rectangles	Shift	Draw a rounded square
	V	Constrain height
	H	Constrain width
Ovals	Shift	Draw a circle
	V	Constrain height
	H	Constrain width
Polygons	Shift	Constrain sides to 45° angles

Changing Object Attributes

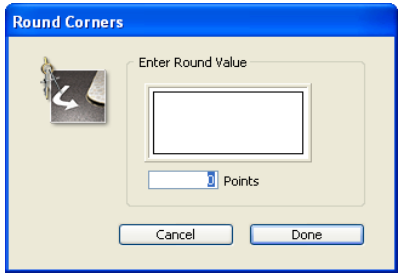
Each of the following object attributes can be changed using commands in the **Object** menu:

- **Fill Pattern**: The pattern displayed inside the border of an object. All objects except lines have fill patterns. The default fill pattern is solid white.
- **Fill Color**: The color displayed inside the border of an object. All objects except lines have fill color. The default fill color is black.
- **Line Pattern**: The pattern of a line or border. The default line pattern is solid.
- **Line Color**: The color of a line or border. The default line color is black.
- **Line Width**: The width of a line or border measured in points. The default line width is 0.25 points (also called a hairline).

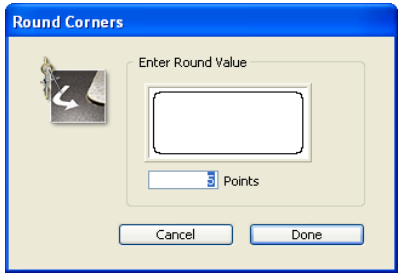
- **Arrowhead**: The arrows that appear on one or both ends of a line; only lines have arrowheads. The default is to have no arrowheads. If the selected object is a line, the **Arrowhead** menu item is enabled. The Arrowhead submenu has the following items.



- **Round Corners**: The amount of rounding of the corner of a rounded rectangle. The default rounding is 1/4 inch. If the selected object is a rectangle that you created with a rectangle tool, the **Round Corners** menu item is enabled. When you choose **Round Corners**, the following dialog box appears:



When you enter a value in the entry area, the preview area shows the effect of the value:



Adding Text

As described in the "Modifying Graph Features" paragraph on page 1092, most of the text that you want to add to a graph — such as axis labels, titles, legends — can be added using menu items in the **Chart** menu.

To add extra text to a graph, you must first create a text object and then enter the text. A text object is a container for text.

► To create a text object:

1 **Select the Text tool** .

The pointer changes to an I-beam .

2 **Hold down the mouse button and drag the mouse to create a rectangular text area.**

3 **Release the mouse button.**

You can also simply click in the window after you have selected the Text tool to create a text object of default size (three inches in width). The default height is determined by the font and font size you have selected.

4 **If you have not already done so, click inside the text object to insert the pointer.**

When you place the pointer in the text object, it becomes an insertion point .

5 **Type your text.**

6 **When you have finished typing the text, select any other tool.**

Unlike other objects, text objects are not selected after you create them.

Once you have created a text object and added text, you can change the text by editing it — for example, copying, cutting, or pasting it. You can also change its attributes, such as its font, size, style, and justification. For more information about changing text attributes, refer to the next section, "Changing Text Attributes".

Changing Text Attributes

Text attributes are applied only to text inside a text object. Each of the following text attributes can be changed in the **Text** menu:

- **Font**: The typeface of the text.
- **Size**: Font size is measured in points. The default size is 12 point.
- **Style**: Styles include plain, bold, and italic. The default style is plain.
- **Color**: The default color is black.
- **Alignment**: Text can be left, right, or center aligned. The default alignment left.

Resizing Objects

You can make an object larger or smaller by resizing it. You can select one or more objects to resize. If you select more than one object, all selected objects are resized by the same amount.

You can resize the height and width at the same time, maintaining the sizing proportion between the two.

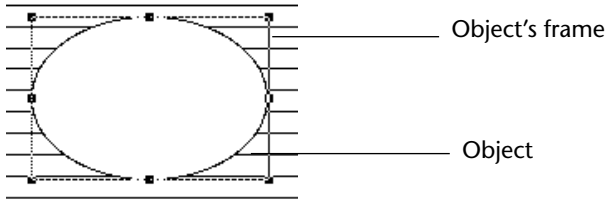
► To resize an object:

1 **Select the object.**

2 **Hold the mouse button down on a selection handle and drag it up, down, or diagonally to change the object's size.**

If you drag a corner, the height and width change. If you drag a side, either the height or the width changes.

To display the object's outline — rather than its rectangular frame — hold down the **Alt** key (**Option** key on Macintosh) while resizing.



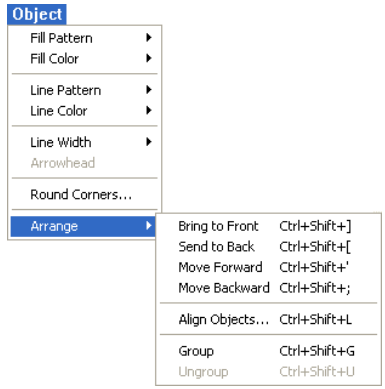
To resize the object so that the height and width retain the same proportion to each other, hold down the **Shift** key while you are resizing.

3 **Release the mouse button when you have finished.**

The object is resized.

Arranging Objects

When you have two or more objects in a 4D Chart document, you can use the items in the **Arrange** submenu of the **Object** menu to arrange them. The **Arrange** submenu is shown below.



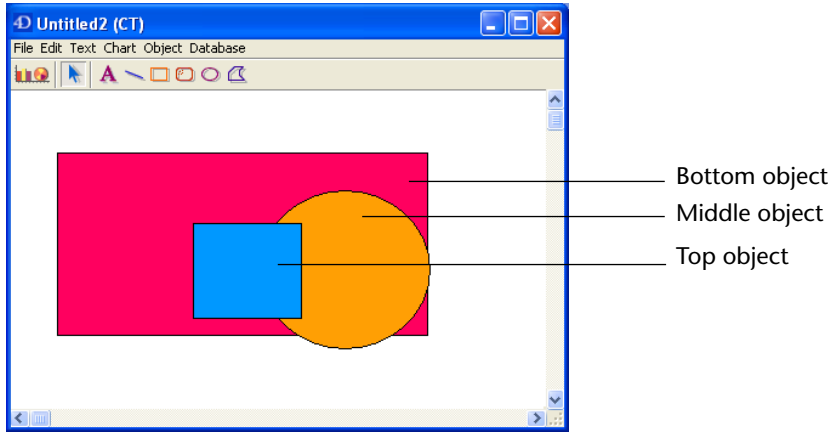
You can choose from the following actions in the **Arrange** submenu:

- Bringing an item to the front
- Sending an item to the back
- Moving an item forward one level
- Moving an item backward one level
- Aligning objects in relation to each other
- Grouping a set of objects
- Ungrouping a set of objects.

The following sections describe these actions in more detail.

Changing the Stacking Order

When you draw objects, they can partially or entirely overlap one another. The order in which objects are placed when they overlap one another is called the *stacking order*. The example below displays several objects and their stacking order:



As you work with objects, you may want to change the stacking order of objects; to do so, you can move one or more objects in front of or behind other objects.

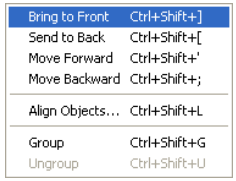
- **Moving an Object in Front of All Others**: By moving an object in front of all others, you move it to the top layer.
- **Moving an Object Behind All Others**: By moving an object behind all other objects, you move it to the bottom layer.
- **Moving One Object in Front of Another**: Moving an object in front of another object allows you to move the selected object one layer higher — that is, place it closer to the top layer.
- **Moving One Object Behind Another**: Moving an object behind another object allows you to move the selected object one layer lower — that is, place it closer to the bottom layer.

► To modify the stacking order:

1 **Use the arrow pointer to select an object.**

2 **Choose a command in the Arrange submenu of the Object menu.**

The **Arrange** submenu is shown below:

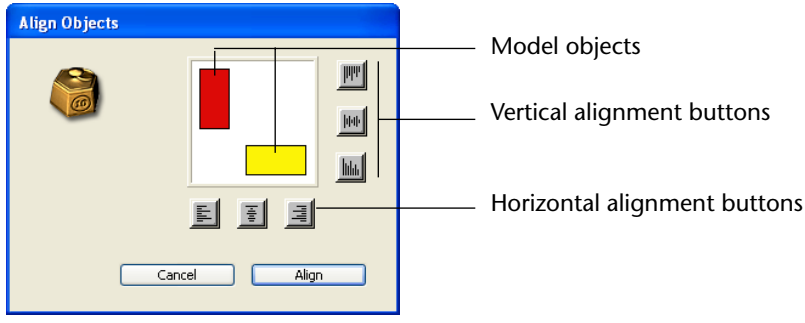


Aligning Objects

When you align objects, you position them in relation to each other. Aligning objects ensures that they are placed precisely in relation to each other. You can align objects both horizontally and vertically.

The object that is farthest in the selected alignment direction is the reference for object alignment. For example, if you want to align the left sides of several selected objects, the selected object that is furthest to the left remains fixed, and the other objects are aligned with it.

When you select **Align Objects** from the **Arrange** submenu, the Align Objects dialog box appears:



Use the following icons to align the selected objects with each other:

Icon	Meaning
	Align the left edges of selected objects.
	Align the centers of selected objects along a vertical axis.
	Align the right edges of selected objects.
	Align the top edges of selected objects.
	Align the centers of selected objects along a horizontal axis.
	Align the bottom edges of selected objects.

Select a maximum of one vertical and one horizontal alignment. The model objects move to indicate the effects of the alignment.

Note To deselect an icon, click it again.

► To align objects:

1 **Select the objects to be aligned.**
Shift-click to select several objects.

2 **Choose Align Objects from the Arrange submenu of the Object menu.**
The Align Objects dialog box appears.

3 **Click the desired alignment icons and click the Align button.**

Grouping and Ungrouping Objects

By *grouping* objects, you can combine several objects into one. A grouped object acts like a single object when you manipulate or edit it. You can work with a grouped object as you would any other object; you can change its attributes, resize it, and so on.

When you group several objects, all attributes of the individual objects are retained. However, if you change any attribute of the group, that change affects all objects in the group. For example, if you select a new fill pattern for the group, it is applied to each object in the group.

Ungrouping an object breaks it into its component objects. When an object is ungrouped, each piece becomes a separate object again. Ungrouped objects retain any changes made to them while part of a group.

► To group objects:

1 **Select all objects to group.**

2 **Choose Group from the Arrange submenu of the Object menu.**
The objects become one object.

You can break a grouped object into its separate objects by ungrouping it.

► To ungroup objects:

1 **Select an object to ungroup.**

2 **Choose Ungroup from the Arrange submenu of the Object menu.**
The object is ungrouped and each object is selected.

Inserting Dynamic 4D References

You can add dynamic references to field values or to 4D expressions by inserting expressions into text objects.

Using field references and 4D expressions, you can create 4D Chart documents that incorporate information from your database. For example, you can use field information from the records to make graph titles. You can use 4D expressions to perform tasks such as computing numeric values or concatenating text information.

In this section, you will find information about the following topics:

- Inserting field values into a 4D Chart document,
- Inserting a 4D expression into a 4D Chart document,
- Displaying field and expression values,
- Formatting field and expression values,
- Changing a dynamic reference to static text.

Understanding Values and References

You can display the information from 4D as either *values* or *references*. A value is the actual information stored in a field or calculated from an expression. A reference is the name of the field or the text of the expression.

When expressions and fields are displayed as references, they are surrounded by the following symbols: « and ».

For example, a reference to the First Name field in the [Stationery Order] table would appear as:

«[Stationery Order]First Name»

4D Chart inserts these symbols when a field is inserted into a text object. When you insert any other type of reference, such as a 4D function or variable, you must indicate that it is a reference. 4D Chart will then add the « and » symbols to differentiate the reference from normal text. For more information on referencing expressions, see the "Inserting 4D Expressions" section on page 1128.

When references are displayed as values, the « and » symbols do not appear; for example, a field value for First Name might be:

James

The field value appears as standard text.

Field references and 4D expressions always refer to the current record and are updated whenever the current record changes. If there is no current record, no value is displayed.

Inserting Field References

By inserting a field reference in a 4D Chart document, you add dynamic information to the document. As the field is updated, so is the value in 4D Chart. You can use fields from any table in the database.

The field reference or value appears in a text object, except for references to picture fields. The text object containing a reference acts as any other text object; you can change its attributes, move it, and so on.

Using Field References in a Plug-in Window

When you insert a field reference in a 4D Chart plug-in window, the value that is displayed when you choose **Show Values** from the **Database** menu is the field's value for the current record. If no record is currently loaded there will be no value displayed in the 4D Chart plug-in window.

In order to show the value stored in a field for a particular record, make sure that the record is loaded — by displaying the record in an input form, for instance. If the current record changes, the value displayed in the field changes.

Using Field References in a 4D Chart Area on a Form

When you insert a field reference in a 4D Chart area on an input form, the value that is displayed when you choose **Show Values** from the **Database** menu is the field's value for the current record.

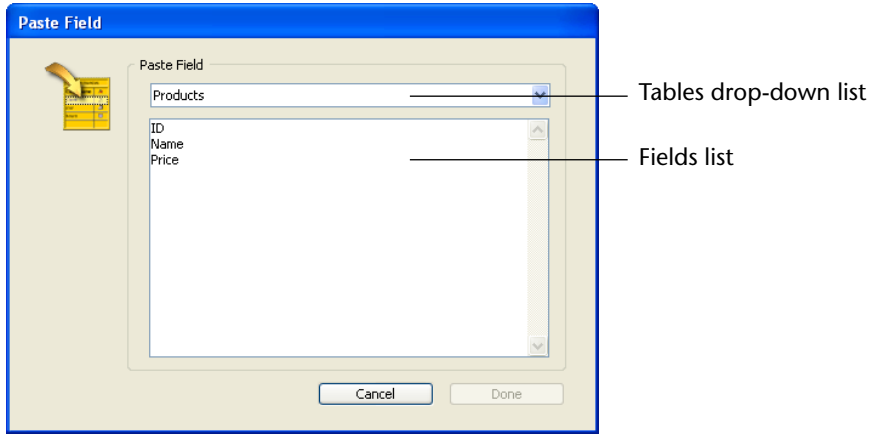
You can use the Paste Field dialog box to paste a field reference into a document.

► To paste a field reference into a 4D Chart document using the Paste Field dialog box:

1 **Click the mouse button where you want to insert the reference.**
If you click inside a text object, you cannot insert a Picture field.

2 **Choose Paste Field from the Database menu.**

The Paste Field dialog box appears. All tables in the database are listed in the Tables drop-down list. The fields for the selected table are displayed in the Fields list.



3 **Select the table from which you wish to paste a field reference from the Tables drop-down list.**

The fields in the selected table are displayed in the Fields list.

4 **Select the field you wish to paste from the Fields list and click OK.**

A reference to the field is pasted into the document in a text object.

You can select fields from a drop-down list containing all available tables and fields. You cannot paste references to Picture fields when you select a field from a drop-down list.

When you are working in a form, you can choose fields from a drop-down list that displays the fields in the current table, or from a drop-down list of fields for all tables in the database.

When you use 4D Chart in a plug-in window, the drop-down list always contains both tables and fields.

► To insert a field into a 4D Chart document using the Tables and Field pop-up menu:

1 **Create a text object where you want to paste the field.**

If you need information about creating text objects, see the "Adding Text" section on page 1118.

2 **Position the I-beam pointer over the text object.**

Make sure that the insertion point is in the text object.

3 **To choose a field from the same table as the form, hold down the Alt key (Windows) or Option key (Macintosh) as you press the mouse button.**

A pop-up menu of fields for the table in which you are working appears. If you are working in a plug-in window, the pop-up menu contains both tables and fields.

4 **To choose a field from another table, hold down the Shift and Alt (Windows) or Option (Macintosh) keys while you press the mouse button.**

If you are working in a plug-in window, you do not need to hold down the **Shift** key.

4D Chart displays a hierarchical pop-up menu of the tables in the database. Each table has a submenu containing its fields.

5 **Select a field.**

A reference to the selected field is pasted into a text object in the 4D Chart document at the insertion point location.

Inserting 4D Expressions

You can insert any valid 4D expression into a document. The expression can be a 4D variable, a 4D function, a plug-in function, or a method that returns a value.

With expressions you can use the power of 4D's language within 4D Chart documents. You can perform calculations, concatenate information from several fields, and so on.

The following table contains some examples of expressions:

Expression	**Comment**
vDate	A variable containing a date
Current date	A 4D function
Current date–vDate	A statement that performs a calculation
DateCalc	A method that returns a value

An expression is evaluated only when you perform one of the following actions:

- Open the document
- Choose **Show Values** from the **Database** menu
- Print the document.

For more information about expressions, refer to the *4D Language Reference* manual.

▶ To insert a 4D expression into a 4D Chart document:

1 **Create a text object where you want to place the expression.**
Since the expression is created from text, you must first create the text object.

2 **Enter the expression text and then select it.**

3 **Choose Reference from the Database menu.**
The text is now enclosed in the « and » symbols, which indicate that it is an expression:

«Current date»

4 **When you have finished, select the Arrow tool.**
To display the value of the reference, choose **Show Values** from the **Database** menu.

Displaying Values or References

When you display values, you show the actual values stored in referenced fields and the values calculated by referenced 4D expressions. For example, if the reference is «Current date», today's date is displayed.

▶ To display values:

- **Choose Show Values from the Database menu.**
The value of each reference is displayed.

Note The values of all references are displayed, regardless of the selected reference.

▶ To display references:

- **Choose Show References from the Database menu.**
The reference for each value is displayed.

Changing the Value in a Reference to Text

You can change the value in a dynamic reference into static text that will not be updated when the value changes.

For instance, you can use the 4D expression «Current date» to display the date you created a graph, and then change that date to a static text object. No matter what the current date is, the text object will always display the creation date of the graph.

Changing a value to text is called *unreferencing* the value, because the value no longer has a dynamic reference to the database. Once you have unreferenced a value, you cannot change it back to a reference.

▶ To change a value in a field reference or expression to text:

1 **Choose Show Values from the Database menu to display all values.**

2 **Select the text block containing the value you want to unreference.**
If the text block contains more than one field or expression, select only the value you want to change.

3 **Choose Unreference from the Database menu.**
The value becomes text.

Changing a Reference to Text

You can change a reference into text by unreferencing it. When you change a reference to text, you change the words of the reference, not the value to which the reference refers.

For instance, if you change the reference «Current date» to text, the text object displays the following:

Current date

When you choose **Show Values** from the **Database** menu, this text does not change, and the current date is not displayed, because the text no longer refers to a 4D expression.

You may want to edit a reference or permanently change it to standard text. If you want to edit a reference, you must unreference it, edit it, and then reference it again.

▶ To unreference a field reference or expression:

1 **Choose Show References from the Database menu to display all references.**

2 **Select the text block containing the reference you want to unreference.**

If the text block contains more than one field or expression, select only the reference you want to change.

3 **Choose Unreference from the Database menu.**

The « and » symbols are removed, and the reference becomes text.

Formatting References

You can use a display format for the value of any numeric, date, or time field, or expression in 4D Chart. For instance, you can use a display format to display dollar amounts using the dollar sign ($), commas, and decimals.

The following table shows some examples of display formats:

Data in Default Format	Display Format	Data in Display Format
3400	$###,##.00	$3,400.00
3/4/07	Month Date, Year	March 4, 2007

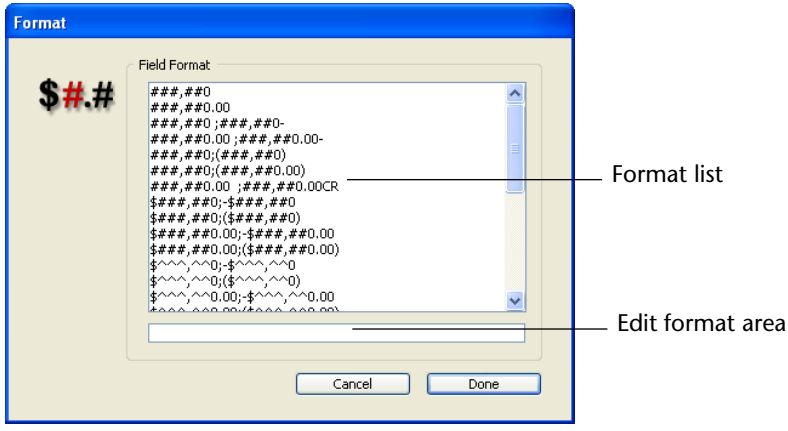
► To format a reference:

1 **Select the reference.**

Since the entire expression is a single value, you can select it by clicking it with the Text tool.

2 **Choose Format from the Database menu.**

The Field Format dialog box appears.



3 **Select an appropriate format from the list.**

The format appears in the text box below the list of formats.

If you want, you can edit a number format or enter a new number format in the Format text box. You cannot edit date or time formats.

For more information about display formats, refer to the "Display Formats" paragraph on page 521.

4 **Click OK.**

The format is listed after the reference. For instance:

«Current date; Abbr.: Month Day, Year»

When you choose **Show Values**, the value of the reference is displayed using the format.

Printing 4D Chart Documents

You can print any 4D Chart document. This section discusses printing 4D Chart documents in the following ways:

- As a single 4D Chart document
- As part of a 4D form
- As a part of a print merge, in which you print a 4D Chart document for each record in a selection of records.

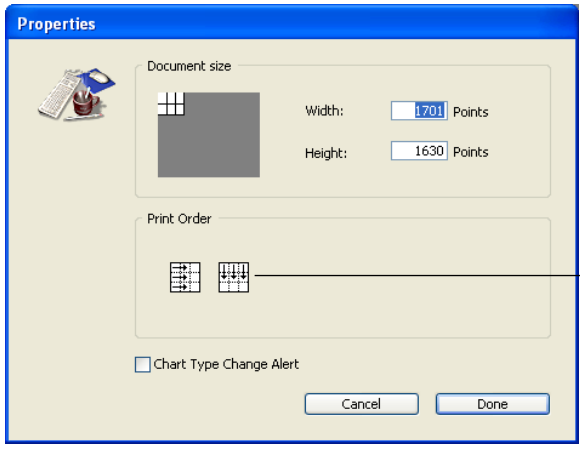
Setting the Print Order

The order in which a multi-page document is printed (either horizontally or vertically) is determined by the print order selected in the Properties dialog box. You can select printing by row  , or printing by column  . The print order affects only the order in which the document prints; it does not affect the page orientation. The default is to print pages by row.

▶ To set the order in which a multi-page document is printed:

1 **Choose Properties from the 4D Chart Edit menu.**

The Properties dialog box appears:



2 **Click the desired Print Order icon.**

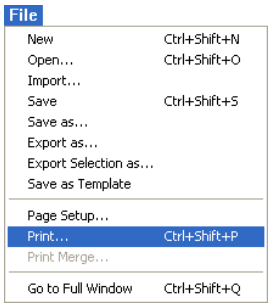
3 **Click OK.**

Printing the Document

You can print 4D Chart documents from a plug-in window or from a 4D Chart area in a form. If the document is more than one page in length, you can select the pages you want to print.

► To print a document:

1 **Choose Print from the 4D Chart File menu.**



The Print dialog box appears.

2 **Select options as necessary.**

3 **Click the Print button to begin printing.**

Printing a 4D Chart Area as Part of a Form

If a 4D Chart area is on a 4D form, you can print it with the record. In this case, you print from 4D, rather than from 4D Chart.

Before printing, be sure to select the records you want to print. For more information about selecting records in 4D, see the "Selecting and Searching Records" paragraph on page 950.

▶ To print a 4D Chart document as part of a record:

1. **Choose Print from the 4D File menu.**
   A dialog box appears so that you can choose the print form.
2. **Choose the print form.**
3. **Click OK.**
   The Page Setup dialog box appears.
4. **Select options from the Page Setup dialog box as necessary.**
5. **Click OK.**
   The Print dialog box appears.
6. **Select options as necessary.**
7. **Click the Print button when you want to begin printing.**
   The selected records are printed, including the 4D Chart documents.

Creating a Print Merge

You can perform a print merge of a 4D Chart document. Performing a print merge allows you to print a 4D Chart document for each record in a selection of records.

The values in any graphs in the document are not updated for each record.

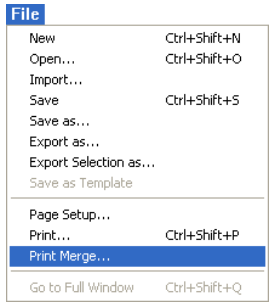
Print merges are most useful when performed from 4D Chart plug-in windows.

You can only perform print merges for documents that contain references to 4D fields. The value in a field reference is determined by the current record being printed. The advantage to performing a print merge is that you can print a document for an entire selection of records without having to load and print for each record separately.

Note For information on adding field references, see the "Inserting Field References" section on page 1126.

► To perform a print merge:

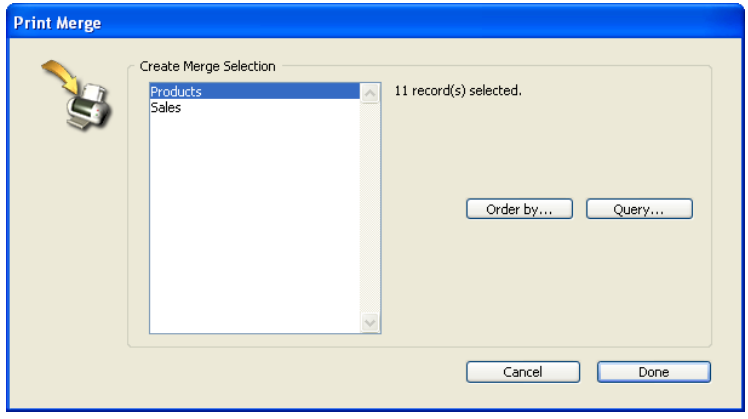
1 Choose Print Merge from the 4D Chart File menu.



The Create Merge Selection dialog box appears.

2 Choose the table whose records should be included in the print merge.

The dialog box displays the number of records currently selected for that table.



4D Chart will only print the document for the selected records.

3 If you want to change the selection of records, click the Query button.

The 4D Query editor appears. After you perform the query in the Query editor, you will return to the Create Merge Selection dialog box. For more information about the 4D Query editor, see the "Standard Searches" paragraph on page 960.

4 **If you want the selected records to be sorted, click the Order by button.**

The 4D Order By editor appears. After you sort the records, you will return to the Create Merge Selection dialog box. For more information about the Order By editor, see the "Sorting Records" paragraph on page 987.

When you have selected the appropriate records, you can print them.

5 **Click OK in the Create Merge Selection dialog box.**

The Print dialog box appears.

6 **Select the appropriate options and click OK.**

4D Chart prints the document for each record in the selection.

22 Publishing or Using Web Services

Overview

4D includes features that enable the publishing or use of Web Services within your database.

What are Web Services?

A Web Service is a set of functions grouped together as an entity and published on a network. These functions can be called and used by any application compatible with Web Services and connected to the same network. Naturally, Web Services are intended to be used to their fullest in the context of publishing on the Internet.

Web Services can carry out all types of tasks, such as supervising the routing of packages at a transporter's, e-commerce, monitoring market values, and so on.

The program publishing the services is called the "server." Any application compatible with Web Services can thus use one or more of these functions; this is the "client" program.
The advantage of Web Services is their interoperability with different information systems: it is not necessary for the server and client programs to be mutually compatible in order for the system to work. From the client application point of view, a Web Service is a "black box": values are sent to it and other values resulting from processing are returned.

The Web Services proposed by the server can be either public or private. There are a great number of public Web Services on the Internet that any application can solicit free of charge.

Maintained by the W3C (World Wide Web Consortium, the regulating authority of the Internet) and major firms of the computer industry, Web Services represent a reliable, lasting and upgradable connectivity solution.

Operation of Web Services — Main Definitions

Web Services transit essentially using the HTTP transport protocol.

SOAP: Web Services use an "open" high-level communication protocol named SOAP (Simple Object Access Protocol). This protocol is based entirely on the XML language, both at the level of the message structure (envelope) and that of the exchanged data. The operation of this protocol is defined by the RFCs (Request for Comment, documents standardizing the various aspects of the Internet), which guarantee it widespread compatibility.

The operation of a Web Service is as follows: A Web Service client sends a request in XML to the server via the SOAP protocol. The server analyzes the request, carries out the requested operation, and returns its response using the same protocol and language.

WSDL: The servers of Web Services generally publish a WSDL (Web Service Description Language) in order to define access specifications for the services being offered. The WSDL enables servers of Web Services to publish the "operating instructions" of the services offered (URLs, lists of methods, parameters, etc.) and comes in the form of an XML file, generally created by the server application itself. This file is not mandatory.

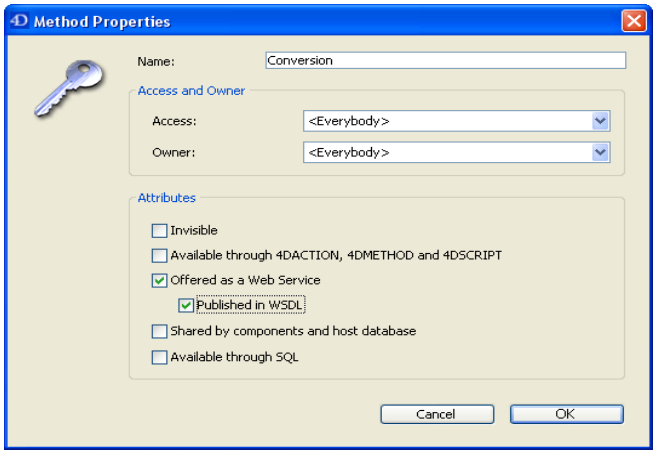
UDDI: The UDDI (Universal Description Discovery and Integration) is a worldwide database that inventories all the public Web Services. Note that it is not mandatory to make a Web Service public and in most cases this will not be necessary.
For more information about the UDDI, please refer to the following address: http://www.uddi.org.

Integration of Web Services in 4D

4D can be used as a Web Services server and/or client. Integration of Web Services into 4D is simple and secure: several configurations enable precise monitoring of publication and subscription conditions.

4D as a Web Services Server

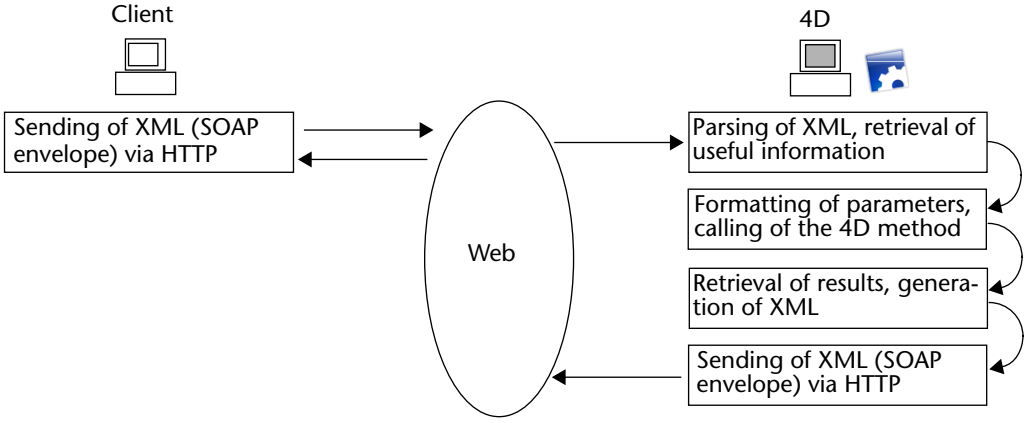
You can decide to publish any project method as a Web Service, without any major modification. Publication is a method property:



Note The publication of Web Services with 4D requires a specific license: the SOAP license.

The 4D Web server automatically handles the management of the service as well as the publication and maintenance of the WSDL file. Parsing the XML content of requests, parameter formatting, sending of results, etc., are performed by 4D without any specific programming being necessary.

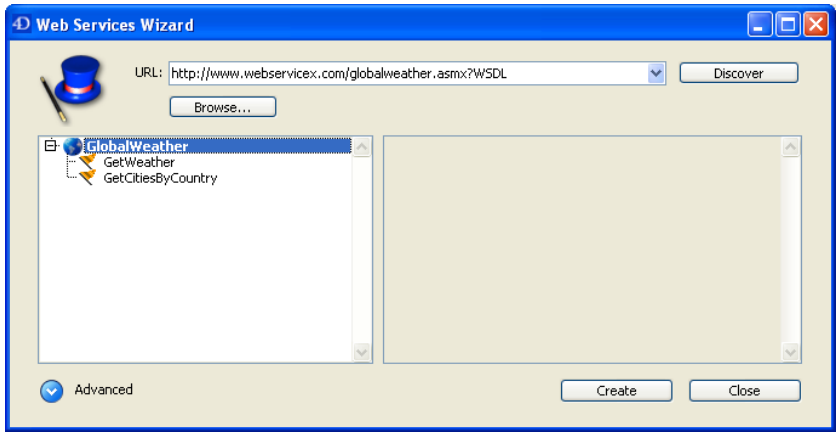
Processing of SOAP requests by 4D



However, if you want to customize request processing, you can use the specific 4D language commands — refer to the “Web Services (Server)” section in the *4D Language Reference* manual.

4D as a Web Services Client

Your databases can use any type of Web Service offered on the Internet or on your network. Most of the time, the Web Services Wizard will enable you to use any Web Service instantly, with a minimum of programming:



Using a Web Service in 4D consists in sending requests over the network and retrieving a response. “Proxy” methods are in charge of these operations. The creation of proxy methods for calling Web Services is entirely automatic and can be performed without programming. Simply call these methods in your code.

It is possible to customize these methods using 4D language commands, in the same way as for the server part.

Security of Web Services

Web Services published by 4D inherit security mechanisms set up for the 4D Web server. Web Services requests thus benefit from the same configurations as conventional Web requests: passwords, On Web Authentication and On Web Connection database methods, use of the SSL protocol, etc.
In addition, specific configurations (for example, Get SOAP info and Is SOAP request commands) allow precise control of Web Service publication.

On the client side, connection to Web Services servers can be carried out in secure mode using SSL. A specific command also allows connection to servers requiring authentication.

Compatibility of RPC, DOC and Complex Types

The communication layer of Web Services (ensuring transport, calling of services and security of exchanges) can operate in two different modes: **RPC** (Remote Procedure Call) mode and **DOC** (Message/Document) mode. The difference between these modes lies at the building level of requests and responses for the server and client. The DOC mode is required by certain client applications.

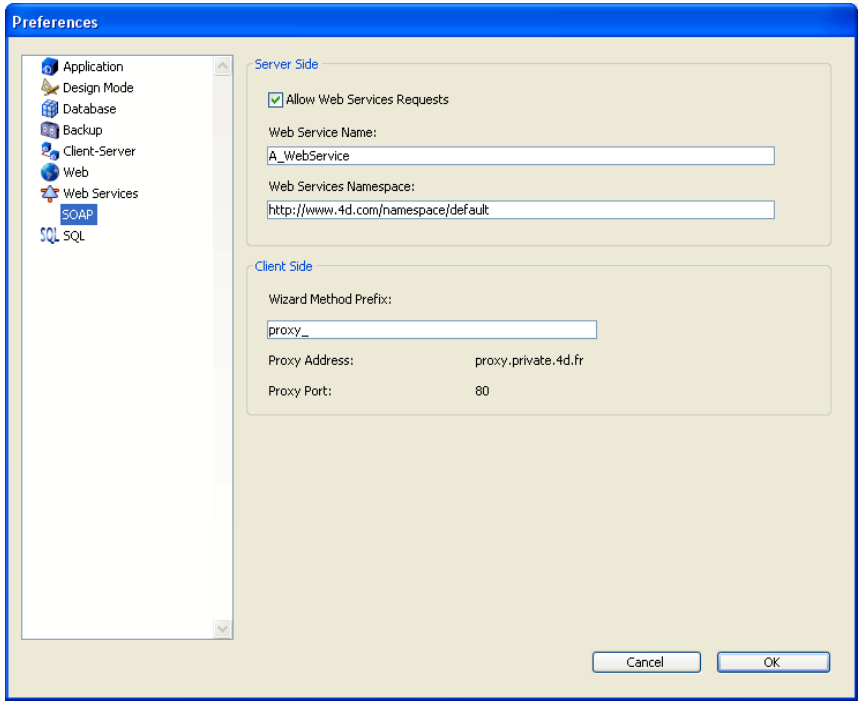
4D supports the RPC and the DOC mode:

- Client side, this support is transparent via the Web Services Wizard. The code generated is automatically adapted to the publication mode.
- Server side, this support is also transparent: the methods are automatically published as Web Services in DOC mode and in RPC mode. The mode choice is carried out via the URL of the WSDL (see the "Generation of the WSDL" paragraph on page 1145).
  - To publish a 4D Web Service in RPC mode, you just need to use the following URL for the WSDL:
    http://ServerAddress/4DWSDL
    The 4D server handles the processing of the requests and responses in RPC. RPC requests are automatically sent to the following address: http://ServerAddress/4DSOAP.
  - To publish a 4D Web Service in DOC mode, you simply need to use the following URL for the WSDL:
    http://ServerAddress/4DWSDL/DOC
    The 4D server handles the processing of the requests and responses in DOC. DOC requests are automatically sent to the following address http://ServerAddress/4DSOAP/DOC.

Two different types of XML data are exchanged via the SOAP protocol: **simple** types and **complex** types. The data of Web Services published in RPC mode can be of either type. Conversely, the data of Web Services published in DOC mode are systematically of the complex type. 4D supports Web Services using simple and complex type data. Complex type XML data cannot be used directly in a 4D database and require specific processing. In most of these cases, the Web Services Wizard will carry out this processing for you; however, it may sometimes be necessary to complete this processing using 4D XML language commands. For more information about this, refer to the "Processing Complex Types" paragraph on page 1159.

Configuration

The **SOAP** page of the database Preferences (“Web Services” theme) can be used to set the general parameters concerning the publication and use of Web Services:



The action of these parameters are described in the following pages. For a more detailed description of each option, refer to the “SOAP Page” paragraph on page 232.

Publishing a Web Service with 4D

Publication of a Web Service in 4D is generally carried out in three stages:

1 **Creation of the method to be published,**

2 **Configuration of the publication (WSDL),**

3 **Publication.**

Additional customizing stages can be defined, but they are not mandatory.

Creating a Web Service Method

You can create any type of project method intended for publication as a Web Service. The method must accept parameters and return a result. It is imperative that these parameters be declared in the method header using commands of the "Compiler" theme.

By default, 4D formats the parameters necessary for the operation of methods when published as Web Services. You can, however, modify these parameters using the SOAP DECLARATION command.

4D automatically takes care of decoding and encoding the data received and sent via SOAP.

Warning: The names of methods are used as XML tags in SOAP requests. In conformity with the XML standard, these names must not begin with a number nor contain spaces. Moreover, in order to avoid any risk of incompatibility, it is advisable not to use any extended characters (such as accented characters).

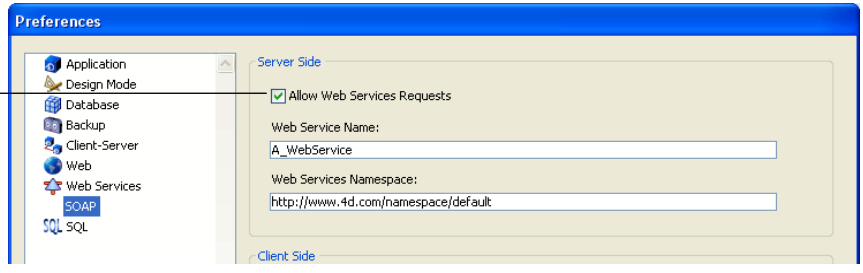
To define and monitor the development of a method published as a Web Service, you must use the commands of the "Web Services" theme — refer to the "Web Services (Server)" section in the *4D Language Reference* manual.

Publication of Methods

To be able to publish one or more methods of your database as Web Services, the following conditions must be met:

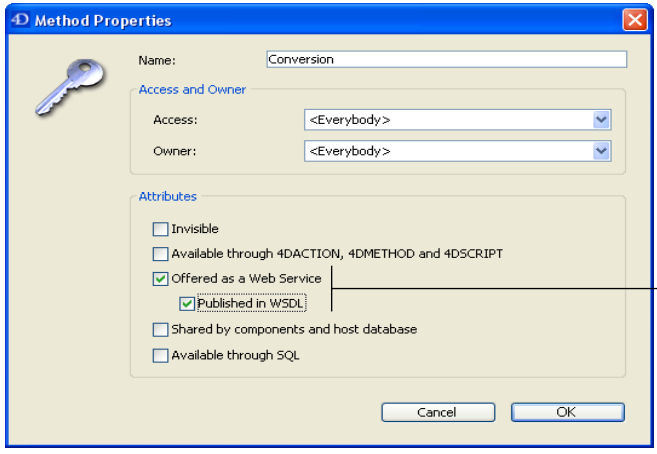
- The machine used as SOAP server (4D single-user application or 4D Server) has a 4D SOAP license.
- The 4D Web server must be launched.
- The **Allow Web Services Requests** option on the **SOAP** page in the Preferences dialog box of the application must be checked. If it is not, 4D refuses SOAP requests and does not generate a WSDL.

Option for publication of Web Services



When this option is checked, 4D creates the WSDL file (see the "Generation of the WSDL" paragraph on page 1145).

- Each method to be published must be **Offered as a Web Service**. This configuration is carried out using an option located in the Method Properties window:

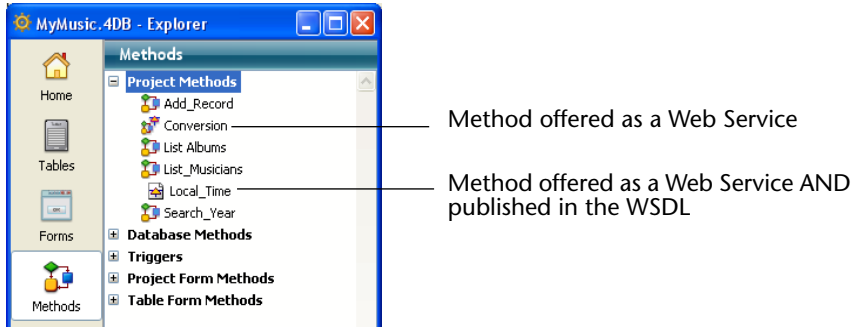


Note For more information about this window, refer to the "Defining the Properties of Project Methods" paragraph on page 773.

When the **Offered as a Web Service** option is checked, the corresponding method can be called as a Web Service via a SOAP request.

Note If the **Published in WSDL** option is also checked, the method will appear in the WSDL of the server (see the following paragraph).

In the 4D Explorer, a specific icon indicates the methods offered as Web Services (as well as those published in the WSDL file):



Generation of the WSDL

The WSDL describes, in XML language, the syntax and information needed for calling the 4D method (Method name, URL, parameters, etc.).

Example of the WSDL file displayed in a Web browser

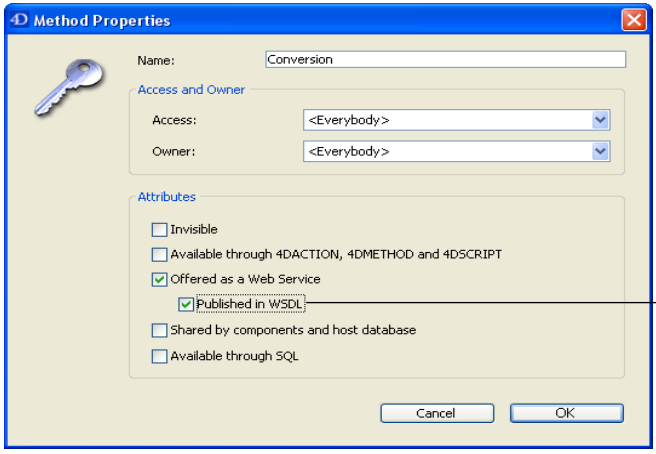
```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- this WSDL file was automatically generated by 4D -->
- <definitions name="tt4d_WebService" targetNamespace="http://www.4d.com/namespace/default"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.4d.com/namespace/default" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
 - <types>
  - <schema targetNamespace="http://www.4d.com/namespace/default" xmlns="http://www.w3.org/2001/XMLSchema"
     xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
   - <complexType name="ArrayOfstring" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    - <complexContent>
     - <restriction base="SOAP-ENC:Array">
        <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="xsd:string[]" />
       </restriction>
      </complexContent>
     </complexType>
    </schema>
  </types>
 - <message name="WS_EuroConverterRequest">
    <part name="Currency_in" type="xsd:float" />
    <part name="From" type="xsd:string" />
    <part name="To" type="xsd:string" />
  </message>
 - <message name="InfoServerRequest">
    <part name="Selector" type="xsd:int" />
  </message>
```

In 4D, the WSDL corresponds to a single Web Service. It defines the methods and their parameters and can be consulted at a specific location. In 4D, the WSDL is not an actual “file” (it only exists in memory and is not written to disk); it is a URL named **4DWSDL** for Web Services published in RPC mode or **4DWSDL/DOC** for Web Services published in DOC mode. It is always located at the root of the Web server. For example, if the address of your Web server is http://www.myserver.com, you could consult the WSDL at the following URL:
http://www.myserver.com/4DWSDL(RPC mode)
http://www.myserver.com/4DWSDL/DOC (DOC mode).

To add or remove a method in the WSDL, simply check or uncheck the corresponding option in the Method Properties window. 4D will immediately update the WSDL contents.

Note In the 4D Explorer, a specific icon distinguishes methods that are published in the WSDL (see the previous paragraph).

When Web Services requests are allowed, 4D automatically and dynamically generates the WSDL of the 4D Web server if at least one method has the **Published in WSDL** option checked in the Method Properties window:

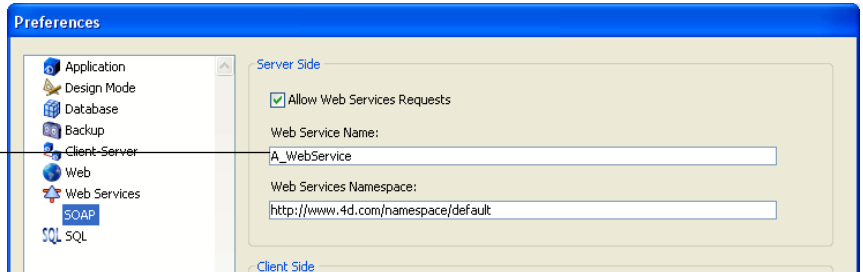


By default, this option is not checked.

Customizing a Web Service Name

Each Web Service published on the Internet has a name. This name is used to differentiate the services both at the SOAP server level (when the server publishes several different Web Services), as well as in the Web Services directories.

By default, 4D uses the name A_WebService. This parameter can be modified on the **SOAP** page of the "Web Services" theme in the database Preferences:

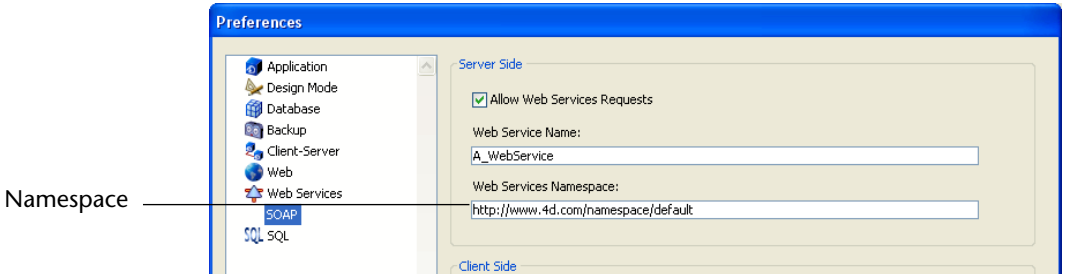


Warning: In conformity with the XML standard for tag names, the character strings used must not begin with a number nor contain spaces. Moreover, in order to avoid any risk of incompatibility, it is advisable not to use any extended characters (such as accented characters).

Customizing a Namespace

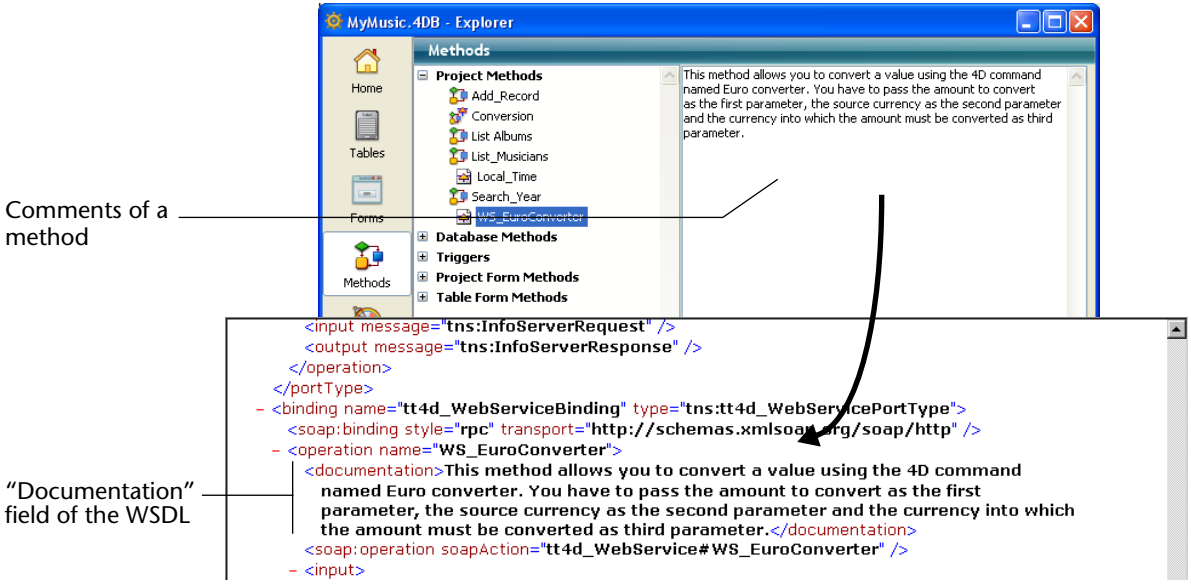
Each Web Service published on the Internet must be unique. The uniqueness of the names of Web Services is ensured using XML namespaces. A namespace is an arbitrary character string used to identify a set of XML tags in a unique way. Usually, the namespace begins with the URL of the company (http://mycompany.com/mynamespace). In this case, it is not indispensable to have anything in particular at the defined URL; what matters is that the character string used is unique.

By default, 4D uses the following namespace: http://www.4d.com/namespace/default. This parameter can be modified on the **SOAP** page of the "Web Services" theme in the database Preferences:



Adding Comments to Published Methods

Any comments associated with methods offered as Web Services and published in the WSDL automatically appear in this file as a "documentation" field.



This mechanism is used to describe or document the published methods. The interpretation and handling of this field will depend on the implementation of the client Web Service.

Accessing a Web Service Published by 4D

Once your Web Service has been published by 4D, any client application that supports Web Services can connect to it. The access mode and the processing of the information exchanged with the Web Service server will depend on the client application used for the operation.

All information needed for the use of a Web Service (such as the URL of the service, the parameters to be used, etc.) are published in the WSDL of 4D. In theory, the use of a Web Service should thus begin with the reading of the WSDL of the SOAP server in order to retrieve this information. In 4D, the URL of the WSDL is http://ServerAddress/4DWSDL (RPC mode) or http://ServerAddress/4DWSDL/DOC (DOC mode).

However, this step is not mandatory. Connection to the SOAP server can be carried out directly.

Here is a list of the values needed to establish SOAP requests, as well as their method of definition:

- **Access URL to a Web Service published by 4D**
  http://ServerAddress/4DSOAP/ (RPC mode) or
  http://ServerAddress/4DSOAP/DOC (DOC mode)
  (not customizable).
- **Web Service name**
  By default: A_WebService.
  Customizable value (see the "Customizing a Web Service Name" paragraph on page 1146).
- **Name of published method**
  Name of the 4D project method defined by the developer (see the "Creating a Web Service Method" paragraph on page 1143).
- **Method parameters**
  The parameters must be declared in the method (defined by the developer).
  Default SOAP names: FourD_arg0, FourD_arg1... FourD_argn.
  Customizable names using the SOAP DECLARATION command.

- **Namespace**
  By default: http://www.4d.com/namespace/default.
  Customizable value (see the "Customizing a Namespace" paragraph on page 1147).
- **Contents of SOAP Action field**
  ServiceName#MethodName (not customizable).

Subscribing to a Web Service in 4D

4D allows you to subscribe to Web Services; in other words, to call external Web Services from within your databases.

Using Web Services available on the Internet, you can easily add numerous additional functions to your databases, such as access to stock market information, package delivery follow-up, execution of complex calculations, etc. The multitude of Web Services available on the Internet can fulfill almost every need.

You can also subscribe to Web Services that you have published yourself in other databases and in this way let various 4D databases communicate among themselves.

How it Works

Any 4D database can subscribe to a Web Service; it simply needs to be connected to the Internet.

Generally, to be able to call a Web Service, you must follow the steps described below:

1 **Retrieve the URL of the Web Service to which you want to subscribe.**
To do this, you can use Web sites that inventory Web Services published on the Internet (for example www.xmethods.net) or directories such as the UDDI. In most cases, you must obtain the URL of the WSDL file for the Web Service.

Note 4D can use Web Services published in RPC or DOC mode (see the "Compatibility of RPC, DOC and Complex Types" paragraph on page 1141).

2 **Using the Web Services Wizard, parse the contents of the WSDL of the Web Service to be used and generate the corresponding *proxy method*.**

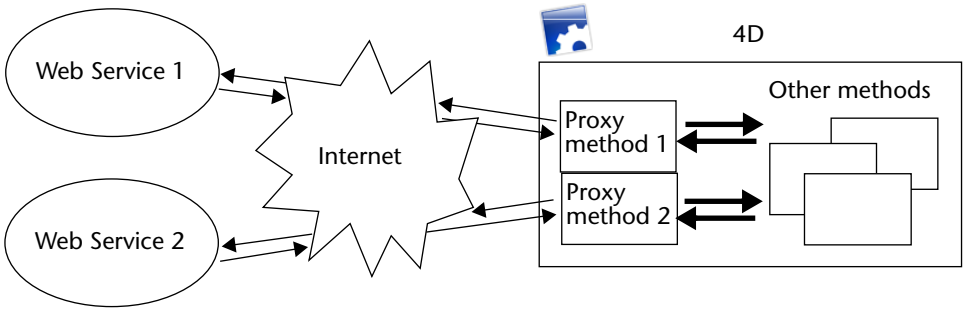
The proxy method is the local project method in charge of interrogating the Web Service and retrieving the returned values. This step is described in the "Using the Web Services Wizard" paragraph on page 1150.

Notes
- It is possible to create proxy methods in the Web Services Wizard without using a WSDL file (simply enter the parameters to be used manually).
- It is also possible to create proxy methods in the Method editor, without using the Web Services Wizard (advanced users).

3 **In the code of your database, call the proxy method each time that you need it by passing the appropriate parameters to it.**
This step is described in the "Calling a Proxy Method" paragraph on page 1158.

The proxy method handles the connection to the Web Service:

Operation of 4D as a client Web Service



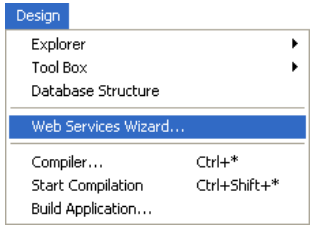
Using the Web Services Wizard

The subscription to a Web Service from a 4D application is handled entirely by the Web Services Wizard. This wizard automatically carries out:

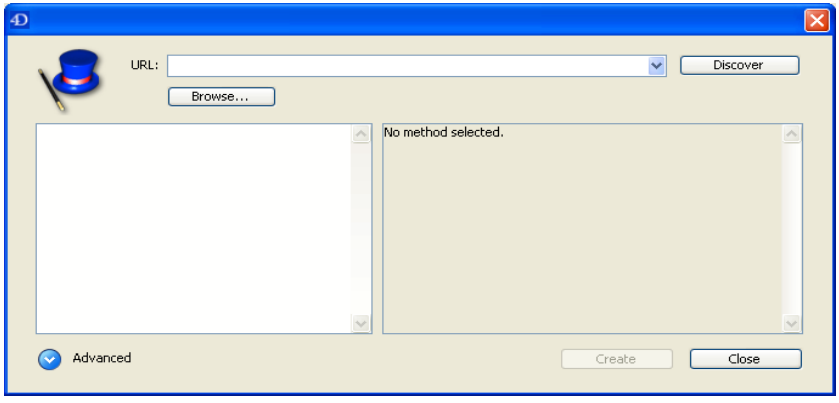
- Parsing of WSDL files for the Web Services to be used,
- Definition of the parameters for the proxy methods to be created,
- Creation of proxy methods.

Wizard Window

To open the Web Services Wizard window, choose the **Web Services Wizard...** command in the **Design** menu of 4D:



The Wizard window appears:



Note It is also possible to display the Web Services Wizard from the options menu of the Methods page in the Explorer (see the "Creating a Method with the Web Services Wizard" paragraph on page 119).

This window includes three areas:

- The "URL:" area allows you to enter or select the URL of the WSDL file for the chosen Web Service. This area is a combo box that stores the previously-entered values in the form of a drop-down list.
- The central area displays the results of parsing the WSDL file contents: names of services and published methods.
- The lower area ("Advanced" parameters, hidden by default) displays the parameters of the method selected in the central area.

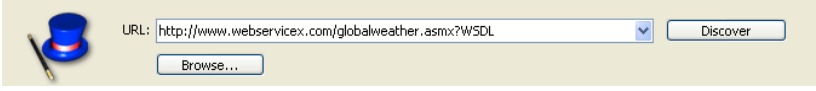
The **Discover** button triggers the parsing of the designated WSDL file and the filling in of the information areas.
The **Browse...** button displays a standard file opening dialog box, allowing the selection of a WSDL stored locally. Its pathname, beginning with "file://", is then displayed in the "URL:" area (it is possible to enter the pathname manually in this area).
The **Create** button is used to generate the proxy method corresponding to the selected Web Service.
The **Close** button closes the Web Services Wizard dialog box.

Parsing of a WSDL and Creation of the Proxy Method (Standard Mode)

Typical use of the Web Services Wizard consists in parsing a WSDL file then generating the corresponding proxy method(s). This standard operation is entirely automatic and does not require any programming or any particular know-how on the part of the user.

► To parse a WSDL file and generate the proxy method:

1 **In the "URL:" area, enter or paste the URL of the WSDL file for the Web Service that you want to use:**

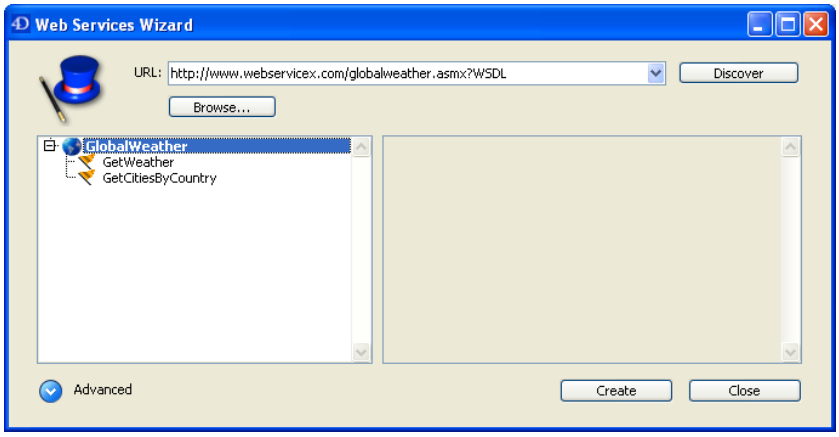


This URL may come, for instance, from a "directory" of Web Services or may have been communicated directly by the server of the Web Service.

You can also specify a local URL, i.e., the address of a WSDL file stored on your hard disk. To do this, click on the **Browse...** button and choose the local WSDL file, or enter its pathname directly in the "URL:" area. The pathname of the local file begins with "file://" then uses the standard system folder separator. You must pass an absolute pathname.

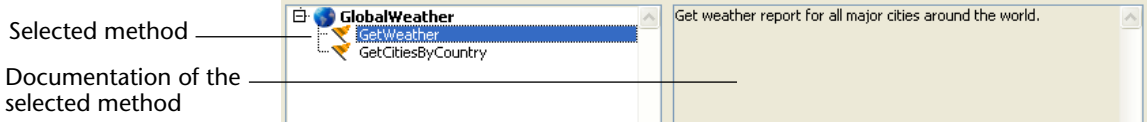
2 **Click on the Discover button in order for 4D to parse the contents of the WSDL file.**

After a few moments, the central area displays the results of file parsing: the name of the Web Service(s) as well as the published method(s) appear in the form of a hierarchical list.

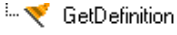


Note You can display the XML source code of the WSDL file directly in your default Web browser by holding down the **Shift** key when you click on the **Discover** button.

Clicking on a Web Service displays its documentation (if any) on the right-hand side of the window. Otherwise, the indication “No documentation” appears.
Similarly, the documentation (if any) for each method appears when you select its name:

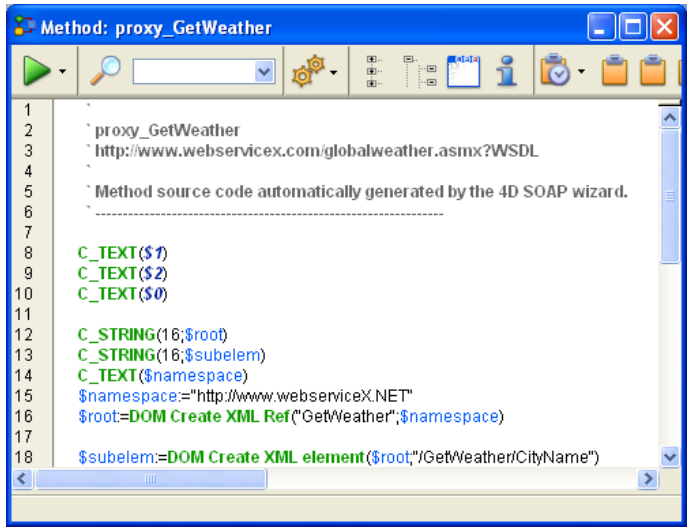


Note If the parsing of the WSDL file reveals the presence of **complex type** parameters, the Wizard displays a yellow flag next to the method concerned.

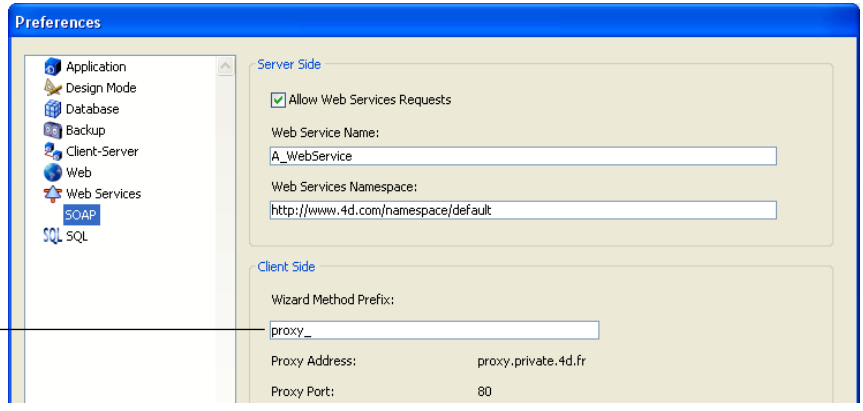


3 Select the Web Service method that you want to use then click on the Create button.

4D instantly generates the corresponding proxy method and displays it in a window of the Method editor:



The name of the proxy method is defined by the concatenation of the default prefix “proxy_” and the name of the Web Service method. The default prefix can be modified in the database Preferences on the **SOAP** page:



Default prefix of proxy method names

The name of the proxy method can also be modified after its creation; this does not influence the operation of the method.

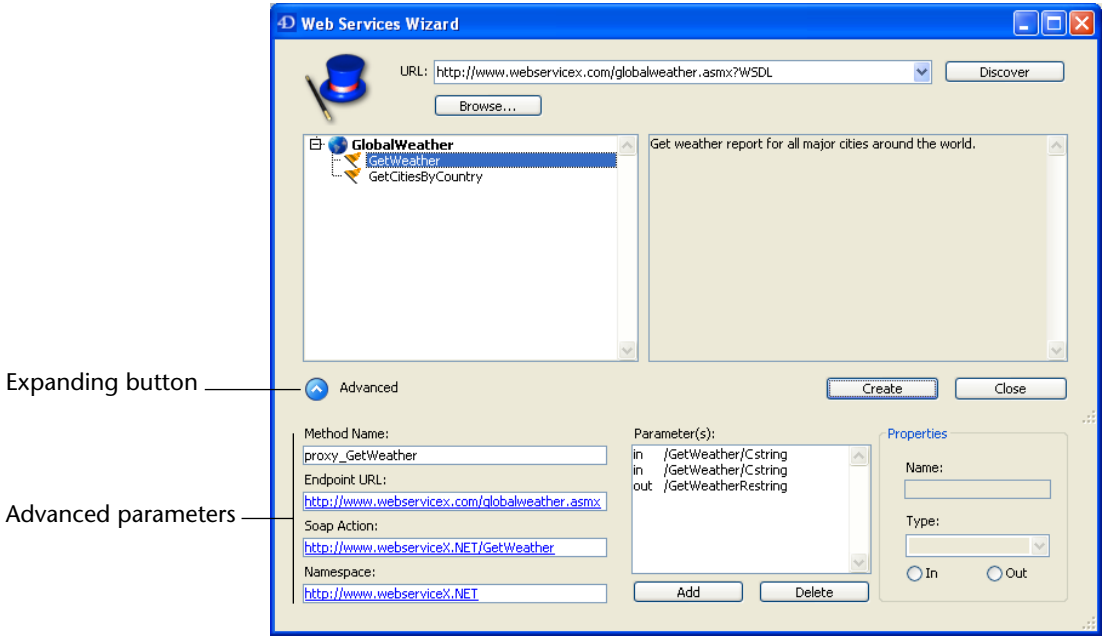
Using Advanced Parameters

Proxy methods generated by the Web Services Wizard from the parsing of a WSDL file are immediately operational and can be used as is (*standard mode*).

However, you might want to modify the parameters resulting from WSDL parsing. For example, it is possible to rename the proxy method.

You can also use the Web Services Wizard to create a proxy method for which you have manually entered the parameters. In this case, do not use the WSDL parser.
It is not mandatory to enter all the parameters to be able to create a method.
It is even possible to not enter any parameters in order to create a proxy method “template” that you can then fill in using 4D programming.

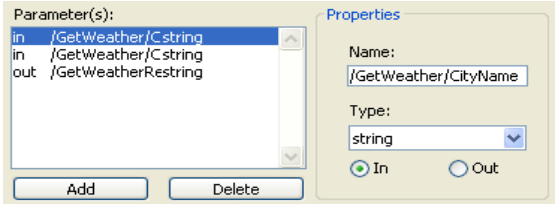
In these non-standard modes, you must use the advanced parameters of the Web Services Wizard. To display these parameters, click on the expanding button located at the bottom left of the Wizard window. The advanced parameter fields then appear. If a method is selected, the fields display its current parameters:



All the parameters are modifiable. Note, however, that modifying parameters stemming from WSDL parsing (except for the method name) must be done with precaution because the operation of the Web Service may be altered as a consequence.

Here is a description of the advanced parameters:

- **Method Name**: Name that the Wizard will give to the proxy method to be created. By default, this name is made up of the prefix "proxy_" (modifiable in the Preferences) followed by the name of the selected method. This name can be modified freely (for instance, if it already exists in the database) without this having any influence on the operation of the Web Service.
- **Endpoint URL**: URL to which the proxy method sends the SOAP requests.
- **Soap Action**: Contents of the SOAPAction field. This field generally contains the value "ServiceName#MethodName."
- **Namespace**: Namespace of the Web Service (for more information, refer to the "Customizing a Namespace" paragraph on page 1147).
- **Parameter table**: This table lists the parameters of the published method.



Each row of the table describes a parameter:

- The first column indicates whether the parameter is of the input ("in") or output ("out") type. This characteristic is evaluated from the point of view of the proxy method, and not that of the published method.
- The second column indicates the name of the parameter.

- The third column indicates the SOAP type of the parameter. Different SOAP types accepted by 4D can be displayed in the Type menu located in the Properties area. The Web Services Wizard will be responsible for associating SOAP types with the corresponding 4D types in the proxy method.
The following table lists the types of SOAP values accepted and the corresponding 4D types:

SOAP Type	Corresponding 4D Type
boolean	Boolean
int	Long Integer
time	Time
float	Real
double	Real
date	Date
string	Text
base64Binary	BLOB
ArrayOfBoolean	Boolean array
ArrayOfInt	LongInt array
ArrayOfTime	LongInt array
ArrayOfFloat	Real array
ArrayOfDate	Date array
ArrayOfString	Text array
AsXML[1]	BLOB

1. The AsXML type is not, strictly speaking, a SOAP type, but it is used for supporting complex XML types (see the "Processing Complex Types" paragraph on page 1159).

The Properties area displays the characteristics of the parameter selected in the table. The Web Services Wizard allows you to modify the existing parameters or add parameters, for instance if the specified WSDL file is not up to date.

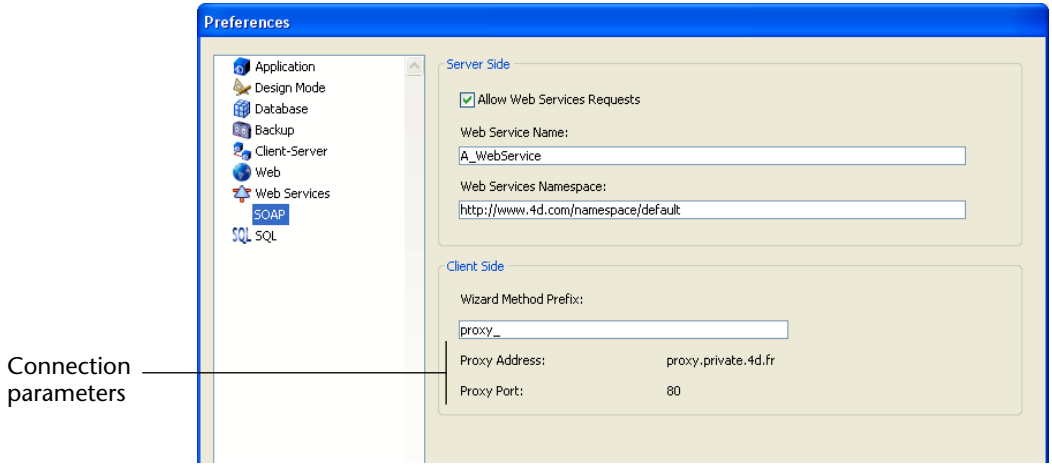
- To modify a parameter, select it and then make your modifications in the Properties area.

- To add a parameter, click on the **Add** button then set its characteristics in the Properties area.
- To delete a parameter, select it in the list then click on the **Delete** button.

Note Modifications made in the advanced parameters will only be taken into account if a proxy method is actually created using the **Create** button.

Display of Connection Parameters

When you subscribe to a Web Service, 4D uses the current Internet connection parameters set for the machine, in particular the Proxy server. You can check these parameters on the **SOAP** page of the "Web Services" theme in the database Preferences:



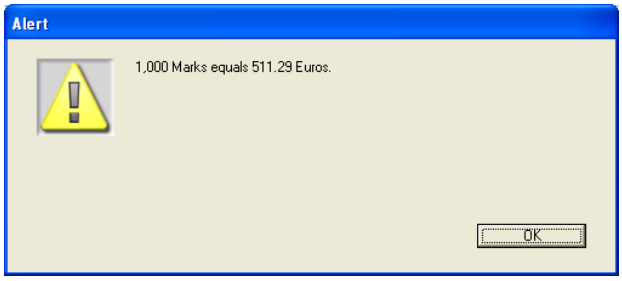
These values are only read by 4D. If you want to modify them, you must do so via the Internet parameters of the machine.

Calling a Proxy Method

To call a proxy method in your code, simply write its name and pass the required parameters to it. These parameters are declared in the header area of the proxy method by the Web Services Wizard. In conformity with the standard syntax for passing parameters between methods in 4D, they are named $0, $1, $2... They can be displayed in the advanced parameters of the description of the published method (see the "Using Advanced Parameters" paragraph on page 1155) and are sometimes described in its documentation.

For instance, in the case of a method named *WS_EuroConverter*, the proxy method could be called in the following manner:

After execution of the method, the following warning is displayed:



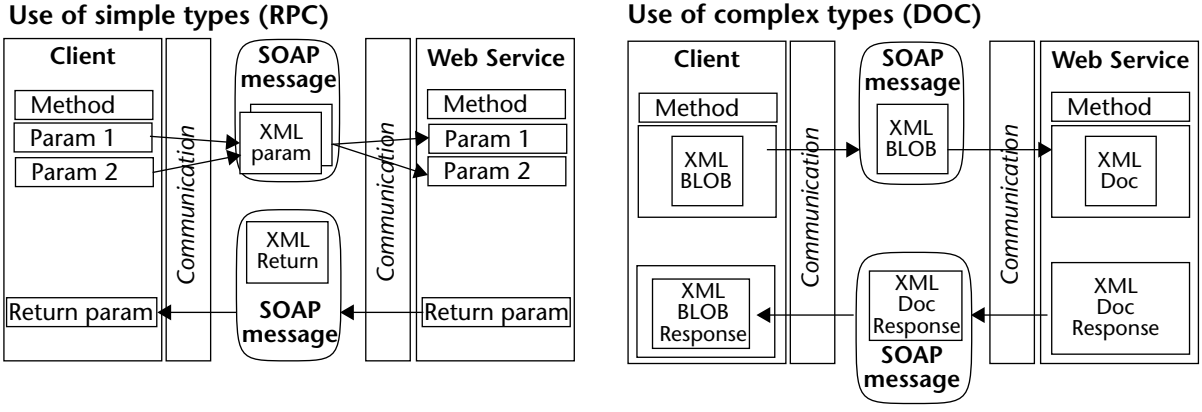
Processing Complex Types

4D enables you to use Web Services published in either RPC or DOC mode and including complex types (see the "Compatibility of RPC, DOC and Complex Types" paragraph on page 1141).

Note Despite the fact that they are complex XML types, data arrays are handled by 4D as simple types.

The proxy methods generated by the Web Services Wizard that include complex types (i.e., published in RPC mode with complex types or in DOC mode) are similar to standard proxy methods. However, you will notice that in certain cases with these Web Services, the CALL WEB SERVICE command includes, as a parameter, a constant containing the word *manual*.

In fact, the use of such Web Services requires additional processing. The main reason for this is that complex types are exchanged in the form of documents or XML *elements*. This means that in order to extract or include information in these SOAP parameters, prior XML parsing is necessary — whereas in the case of simple types, the parameter values are directly readable.



Only arrays and complex type data on one level (a single hierarchical level in the SOAP request) are fully supported by the Web Services Wizard. If more complex elements are found in the request, the Wizard will indicate this by a flag displayed next to the method name. Support of this type of Web Service generally requires custom processing by the developer.

In 4D, complex type parameters (except for arrays) are handled in the form of BLOBs. The XML commands of 4D can be used to process the contents of these BLOBs. For more information, refer to the “Web Services (Client)” and “XML” themes of the *4D Language Reference* manual.

23 Administration of Integrated Servers

4D has two integrated servers: a Web server and an SQL server.

The administration of these servers is very simple and requires only a few settings. There are administration commands that can be used to control their execution. Note that in order to activate these servers, you must have appropriate licenses.

This chapter covers the following subjects:

- Management and administration of the 4D Web server,
- Management and administration of the 4D SQL server.

Note This chapter does not cover the administration of the 4D Sever, the 4D data and application server. 4D Server is covered by its own specific documentation.

Management of 4D Web Server

4D single-user applications and 4D Server include a Web server that can be used to publish the data of your database on the Web, as well as HTML pages, pictures or any type of contents. Any Internet browser can connect to it and use the data.

When the Web server operates in contextual mode, a 4D application can be published “as-is”, so that Web browsers connect in the same manner as in the Application environment.
The 4D Web server automatically manages the conversion of forms into HTML (HyperText Markup Language) format and data integrity.

You can also define custom methods and static or semi-dynamic HTML pages and manage all the data sent and received by the Web server by programming.

The server can also operate in non-contextual mode. In this case, it acts as a standard Web server.

Regardless of the Web server operating mode, you can continue to use your database, for example entering or sorting records, even though the server is launched. You can decide at any time to start or stop the Web server of your application.

How to configure the Web server as well as the basic principles for building an application that is intended to be published on the Web are covered in the "Web Server" chapter of the 4D *Language Reference* manual.

Server Publishing Conditions

In order for the 4D Web server to be operational, the following conditions must be met:

- You must have a "Web Server" license. For more information on this, please contact 4D, Inc.
- Your computer must be connected to a network (Internet or Intranet).
- Under Mac OS X, you must have set up the access to the TCP port for Web server publishing. For more information, refer to the "Web server configuration and connection management" section in the *4D Language Reference* manual.

Starting and Stopping the Web Server

You can start or stop the 4D Web server at any time. These operations can be carried out in the Design environment or via language commands.

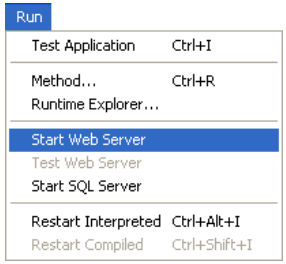
Starting the Web Server

▶ To start the 4D Web server:

1 Check that the publishing conditions are met.

These conditions are described in the paragraph "Server Publishing Conditions" on page 1162.

2 **Choose Start Web Server from the Run menu.** This is a toggle command. When the Web server is already started, it changes to **Stop Web Server**.



OR

In a method, execute the START WEB SERVER command.

For more information, please refer to the "Web Server" chapter of the 4D *Language Reference* manual.

If the network is not available, a warning dialog box will inform you of the fact.

The "Web Server" process, intended to manage Internet and/or Intranet access to the database, is then opened and the database is accessible to browsers.

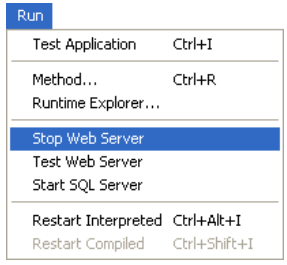
When the Web server is started, 4D automatically ensures the control of data integrity and the locking of records if necessary in the case of simultaneous and multiple accesses to the database (for example, when several different browsers connect to it). In this case, the control of data integrity is identical to that carried out during client/server connections.

While your database is published on the Web, you can continue to work in the Application environment, or continue to develop the database using the Design environment.

Stopping the Web Server

► To stop the 4D Web server:

1 **Choose Stop Web Server from the Run menu.**This is a toggle command. When the Web server is already started, it changes to **Start Web Server**.



OR
In a method, execute the STOP WEB SERVER command.

The Web server is stopped immediately and the "Web Server" process is aborted. Browsers can no longer connect to the database.

Publication at Startup

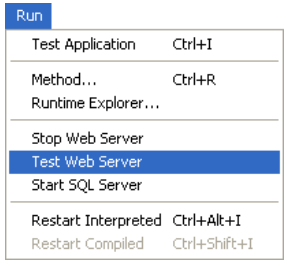
All databases created using 4D can be configured so that they are automatically published on the Web as soon as they are opened by the 4D application.

You can configure these settings in the Design environment, in the Preferences dialog box, **Web/Configuration** page: select **Publish Database at Startup**. For more information on this dialog box, please refer to the paragraph "Configuration Page" on page 218.

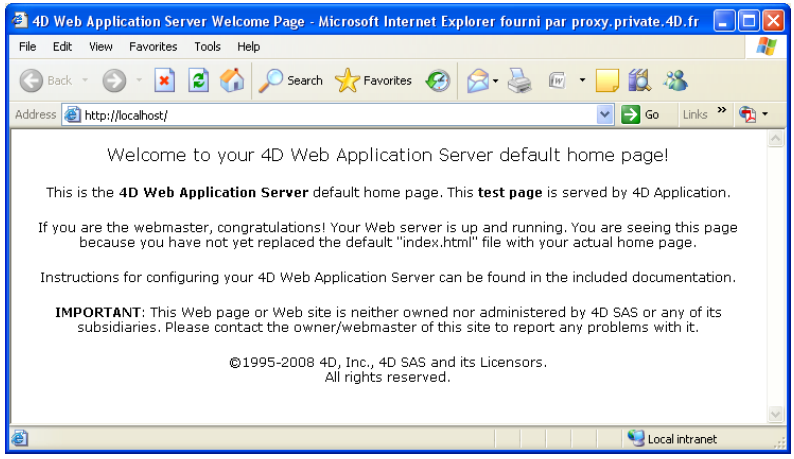
Testing the Web Server

You can check the operation of the 4D Web server at any time from the Design environment.

To do so, you just need to select the **Test Web Server...** command in the **Run** menu of 4D — when the Web server is launched:



When you select this command, the home page of the Web site published by the 4D application is displayed in a window of your default web browser. In non-contextual mode, if you have not modified the default home page, the following page appears:



In contextual mode, the Menu Bar #1 is displayed by default.

This command allows verifying that the Web server, home page display, etc. are working correctly.

The page is called using the URL Localhost, which is the standard shortcut designating the IP address of the machine on which the Web browser is executed. The command takes into account the TCP publication port number specified in the application Preferences.

Management of 4D SQL Server

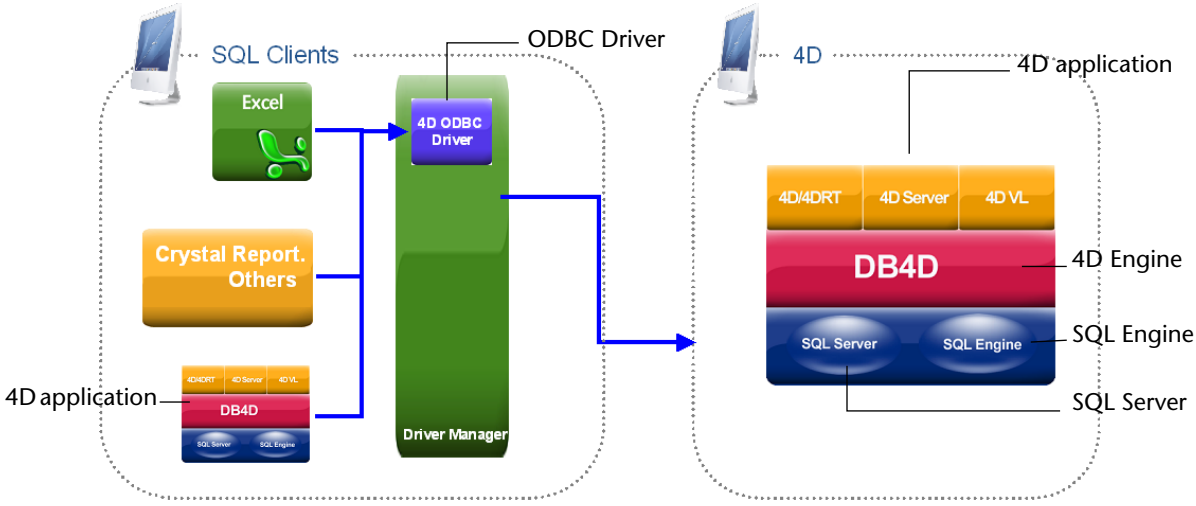
4D v11 includes a powerful SQL server that allows external access to data stored in the 4D database. This access takes place via a 4D ODBC driver.

The SQL server of a 4D application can be stopped or started at any time. Moreover, for performance and security reasons, you can specify the TCP port as well as the listening IP address, and restrict access possibilities to the 4D database.

Note An appropriate license is required to operate a 4D SQL server. For more information, please contact 4D, Inc.

External Access to SQL Server

External access to the 4D SQL server takes place via ODBC. 4D provides an ODBC driver that allows any third-party application (Excel® type spreadsheet, other DBMS, and so on) or another 4D application to connect to the SQL server of 4D. This is summarized in the following diagram:

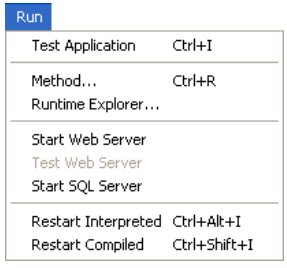


The 4D ODBC driver must be installed on the machine of the SQL Client part. The installation and configuration of the 4D ODBC driver is detailed in a separate manual.

Starting and Stopping the SQL Server

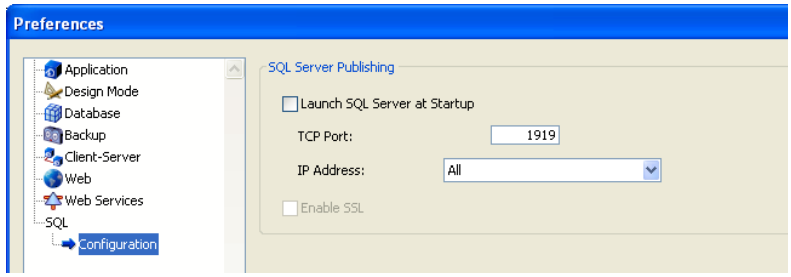
In 4D, the SQL server can be started and stopped in three ways:

- Manually, using the **Start SQL Server/Stop SQL Server** commands in the **Run** menu of 4D:



This is a toggle command. When the server is launched, this menu item changes to **Stop SQL Server.**

- Automatically on startup of the application, via the Preferences. To do this, display the **SQL/Configuration** page and check the **Launch SQL Server at Startup** option:



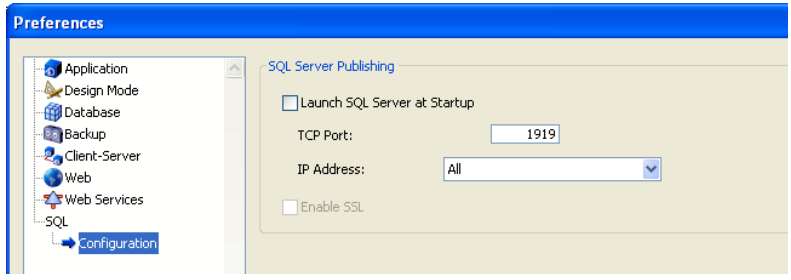
- By programming, using the START SQL SERVER and STOP SQL SERVER commands (“SQL” theme).

  When the SQL server is stopped (or when it has not been started), 4D will not respond to any external SQL queries .

Note Stopping the SQL server does not affect the internal functioning of the 4D SQL engine. The SQL engine is always available for internal queries.

SQL Server Publishing Preferences

It is possible to configure the publishing parameters for the SQL server integrated in 4D. These parameters are found on the **SQL/Configuration** page of the database Preferences:



- The **Launch SQL Server at Startup** option can be used to start the SQL server on application startup (see the paragraph “Starting and Stopping the SQL Server” on page 1166).
- **TCP Port**: By default, the 4D SQL server responds on the TCP port 1919. If this port is already being used by another service or if your connection parameters require another configuration, you can change the TCP port used by the 4D SQL server.

Note If you pass 0, 4D will use the default TCP port number, i.e. 1919.

- **IP Address**: You can set the IP address of the machine on which the SQL server must process SQL queries. By default, the server will respond on all the IP addresses (**All** option).
  The "IP Address" drop-down list automatically contains all the IP addresses present on the machine. When you select a particular address, the server will only respond to queries sent to this address. This is intended for 4D applications hosted on machines having several TCP/IP addresses.

Note On the client side, the IP address and the TCP port of the SQL server to which the application connects must be correctly configured in the ODBC data source definition.

- **Enable SSL**: This option indicates whether the SQL server must enable the SSL protocol for processing SQL connections.

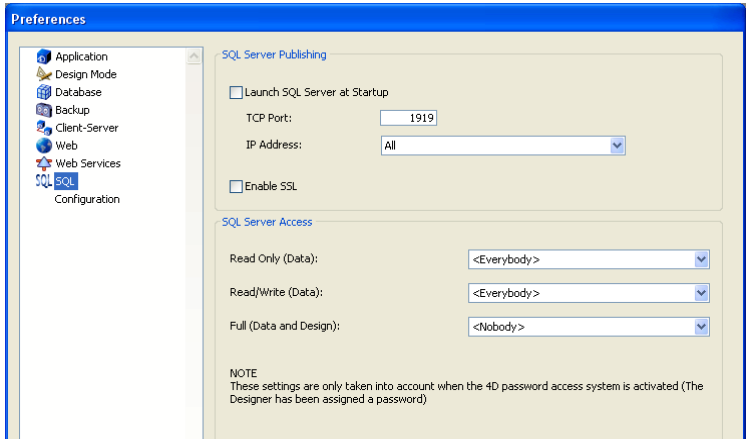
4D Database Access Control

For security reasons, it is possible to limit actions that external queries sent to the SQL server can perform in the 4D database.

This can be done at two levels:

- At the level of the type of action allowed,
- At the level of the user carrying out the query.

These settings are made on the **SQL/Configuration** page of the database Preferences:



For a complete description of these options, please refer to the paragraph "SQL Theme" on page 233.

24 Backup and Restoring of the Database

Introduction

4D includes a full database backup and restore module.

This module allows backing up a database currently in use without having to exit it. Each backup can include the structure file, the data file and any additional files or folders. These parameters are first set in the application Preferences.

Backups can be started manually or automatically at regular intervals without any user intervention. Specific language commands, as well as specific database methods, allow integrating backup functions into a customized interface.

Databases can be restored automatically when a damaged database is opened.

Also, the integrated backup module can take advantage of the log file. This file keeps a record of all operations performed on the data and also ensures total security between two backups. In case of problems with a database in use, any operations missing in the data file are automatically reintegrated the next time the database is opened. You can view the log file contents at any time.

The integrated backup module allows you to:

- Start a complete backup of database files at any time (structure file, data file, log file, attached files, etc.),
- Set up automatic backups at regular intervals — on a hourly, daily, weekly or monthly basis,

- Set advanced parameters for backups (number of sets, file compression, options for startup after a restore, etc.),
- Automatically restore a database and its attached files in case of incident,
- Automatically integrate missing operations stored in the log file into a restored database,
- Roll back operations performed on database data.

Managing Backups

Backing up the database consists of generating a copy of the database and all its necessary files at a given moment. This copy is placed in one or more backup file(s).

Any backup file can later be opened using 4D in case of any incidents that damage the current database; the database will then be restored to its previous state (at the time it was copied).

Performing a Backup

Each backup is performed while taking into account the parameters set in the application Preferences.

The Preferences can be used to define every aspect of the backup:

- Files to include in the backup (data / log, structure, user structure and attached files),
- Location of backup files (main backup file and log backup file).
- Management of log file,
- Backup scheduling,
- Backup options: number and rotation of backup sets; handling of active transactions or index operations; handling of failures; segmentation, compression and integrity of backup files,
- Setting automatic restore options.

These parameters are set at default values corresponding to standard use; changing these values is optional. For more information on these parameters, please refer to the "Configuration of Backup Files" paragraph on page 1176 and the "Backup Settings" paragraph on page 1181.

Starting a Backup

In 4D, a backup can be started in three ways:

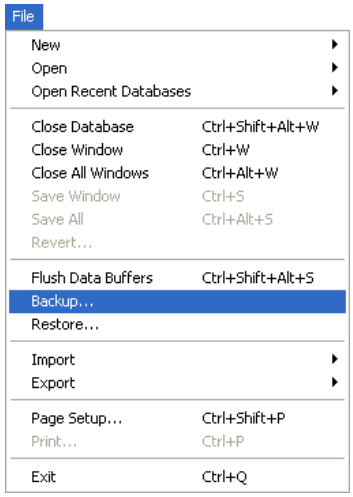
- Manually, using the **Backup...** command in the 4D or the **Backup** button of the Maintenance and Security Center (MSC).
- Automatically, using the scheduler that can be set in the application Preferences.
- Programmatically, using the BACKUP command.

The choice will depend on your use of the database and your backup strategy.

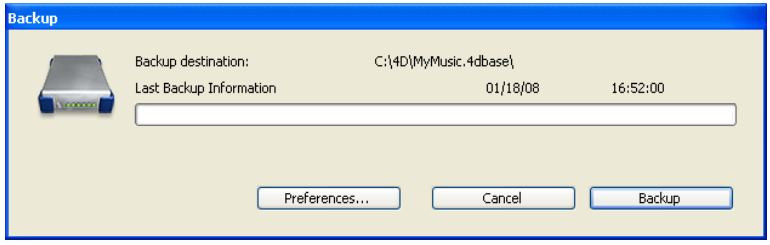
4D Server A backup can be started manually from a remote machine using a method that calls the BACKUP command. The command will be executed, in all cases, on the server.

▶ To perform a manual backup:

1 Select the Backup... command in the 4D File menu:

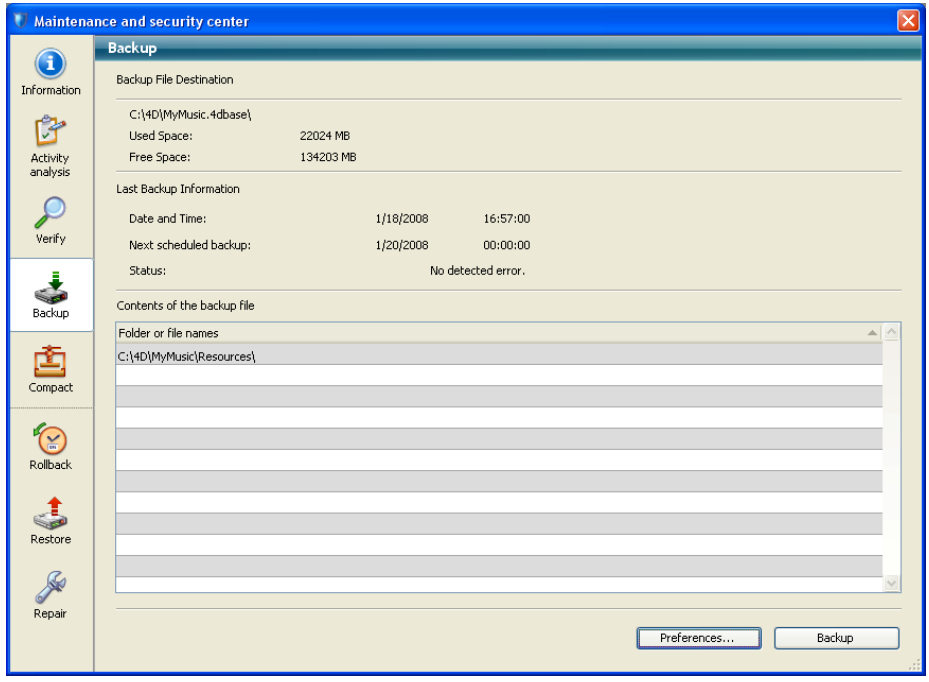


The backup window appears:



OR

Select Maintenance Security Center in the Help menu of 4D and display the "Backup" page.



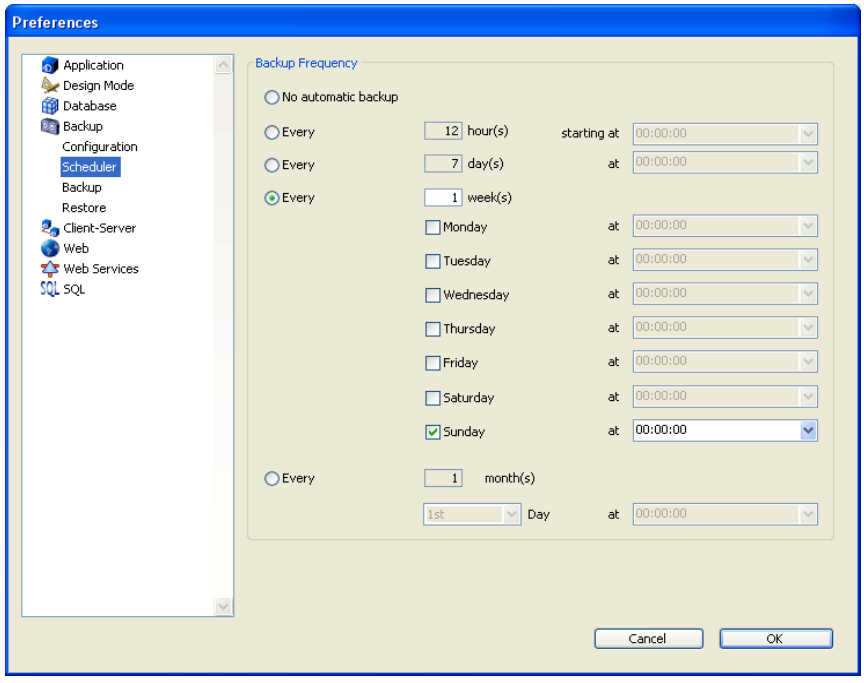
For more information about the MSC, please refer to the chapter "Maintenance and Security Center" on page 1209.

The **Preferences** button causes the general backup preferences to be displayed.

2 **Click Backup to start the backup using current parameters.**

► To perform a scheduled automatic backup:

1 In the Scheduler page of the "Backup" Preferences, set the backup frequency:



Backups are automatically performed at the times defined on this page without any type of user intervention.

Note For more information on using this dialog box, please refer to the "Scheduled Backup Settings" paragraph on page 1185.

► To perform a scheduled backup using 4D language:

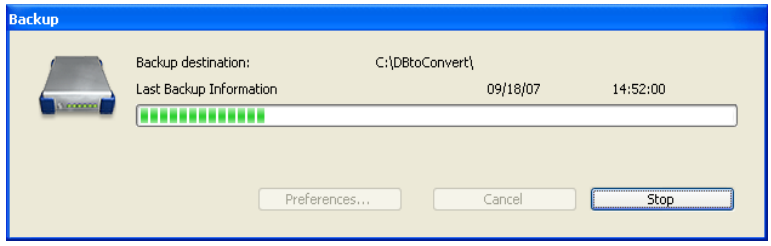
1 Execute the BACKUP command in a method.

The backup starts using the current parameters.

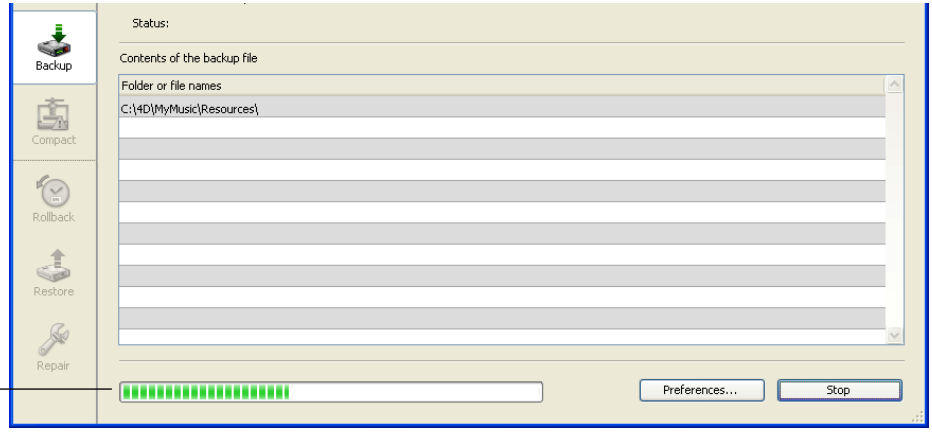
You can use the On Backup Startup and On Backup Shutdown database methods for handling the backup process.
For more information on this, please refer to the *4D Language Reference* manual.

Executing a Backup

Once the backup is started, 4D displays a dialog box with a thermometer indicating the progress of the backup:



This thermometer is also displayed on the "Backup" page of the MSC if you have used this dialog box:



Progress thermometer

The **Stop** button lets the user interrupt the backup at any time (please refer to the "If Backup Fails" paragraph on page 1183).

The result of the last backup (successful or failed) is stored in the "Last Backup Information" area of the **Backup/Configuration** page of the Preferences or in the main window of 4D Server (please refer to the "Last Backup Information" paragraph on page 1180). It is also recorded in the database Backup journal (please refer to the "Backup Journal" paragraph on page 1206).

Accessing the Database During Backup

During a backup, access to the database is restricted by 4D according to the context.

4D locks any processes related to the types of files included in the backup: if only the structure file is being backed up, access to the structure is not possible but access to the data will be allowed.

Conversely, if only the data file is being backed up, access to the structure is still allowed. In this case, the database access possibilities are as follows:

- With the 4D single-user version, the database is locked for both read and write; all processes are frozen. No actions can be performed.
- With 4D Server, the database is only write locked; client machines can view data. If a client machine sends an add, remove or change request to the server, a window appears asking the user to wait until the end of the backup.
  Once the database is saved, the window disappears and the action is performed. To cancel the request in process and not wait for the end of the backup, simply click the **Cancel operation** button.

  However, if the action waiting to be executed comes from a method launched prior to the backup, you should not cancel it because only operations remaining to be performed are cancelled. Also, a partially executed method can cause logical inconsistencies in the database.

Note When the action waiting to be executed comes from a method and the user clicks the **Cancel operation** button, 4D Server returns error -9976 (This command cannot be executed because the database backup is in progress).

Encountering Problems During a Backup

It may happen that a backup is not executed properly. There may be several causes of a failed backup: user interruption, attached file not found, destination disk problems, incomplete transaction, etc. 4D processes the incident according to the cause.

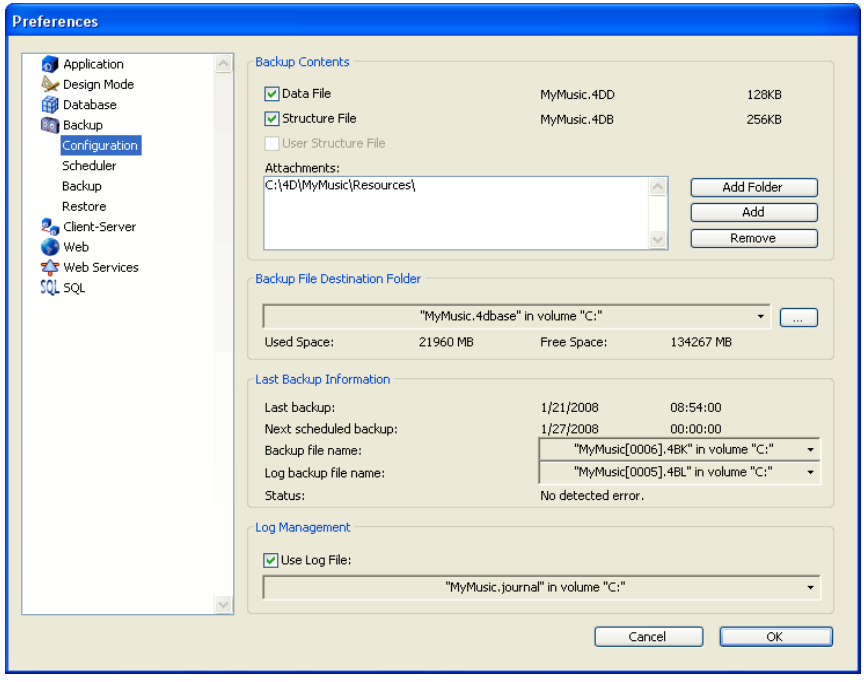
In all cases, the status of the last backup (successful or failed) is displayed on the **Configuration** page in the **Backup** preferences, in the 4D Server window and in the Backup journal (please refer to the "Last Backup Information" paragraph on page 1180).

- **User interruption**: The **Stop** button in the progress dialog box allows users to interrupt the backup at any time. In this case, the copying of elements is stopped and an error is generated. You can intercept this error in the On Backup Shutdown database method.
- **Attached file not found**: When an attached file cannot be found, 4D performs a partial backup (backup of database files and accessible attached files) and returns an error.

- **Backup impossible** (disk is full or write-protected, missing disk, disk failure, incomplete transaction, database not launched at time of scheduled automatic backup, etc.):
  - If this is a first-time error, 4D will then make a second attempt to perform the backup. The wait between the two attempts is defined on the **Backup** page of the Preferences (please refer to the "If Backup Fails" paragraph on page 1183).
  - If the second attempt fails, a system alert dialog box is displayed and an error is generated. You can intercept this error in the On Backup Shutdown database method.

Configuration of Backup Files

The **Configuration** page of the application Preferences lets you set the backup files and their location, as well as that of the log file. It also provides information on the last backup:



These parameters are specific to each database opened by the 4D application.

4D Server These parameters can only be set from the 4D Server machine.

Backup Contents

This area allows you to define which files and/or folders to copy during the next backup.

The upper portion of the area lists the 4D database files and indicates their current size. You must set each file to include in the backup by checking the corresponding option. A dimmed option means that the corresponding file is not available in the database. You can select the 4D files that you want, depending on how often they are updated, their strategic interest, their size, etc. No file is required.
The lower portion lists the file access paths of any attached files in the backup.

- **Data File**: Database data file.
  If the database contains several segments, only the name and size of the first segment are displayed. Of course, all segments are backed up. When this option is checked, the current **log file** of the database, if any, is backed up at the same time as the data. The backup causes the closing and backup of the current log file, then the creation of a new log file. This prevents the size of the log file from becoming excessively large.
  For more information on the log file, please refer to the "Managing the Log File" paragraph on page 1188.
- **Structure File**: Database structure file.
  In cases where databases are compiled and merged with 4D Unlimited Desktop, this option allows you to backup the .exe file (Windows) and the package (Mac OS).
- **User Structure File** (optional): Database User structure file that contains customized user forms (if any). For more information about user forms, please refer to the chapter "User Forms" on page 685.
- **Attachments**: This area allows you to specify a set of files and/or folders to be backed up at the same time as the database. These files can be of any type (documents or plug-in templates, labels, reports, pictures, etc.).
  You can set either individual files or folders whose contents will be fully backed up. Each attached element is listed with its full access path in the "Attachments" area.
  - **Add Folder**: When you click this button, 4D displays a dialog box that allows selecting a folder to add to the backup. In case of a restore, the folder will be recuperated with its internal structure.

You can select any folder or volume connected to the machine, with the exception of the folder containing the database files.

- **Add**: When you click this button, 4D displays a dialog box that allows selecting a file to add to the backup.
  You cannot select a database file as an attached file.
- **Remove**: This button allows you to remove the selected file from the list of attached files.

Note It is possible that one or more attached files are not accessible when the backup is executed (modified name or access path, disk disconnected, etc.). In this case, the backup is executed without the missing file(s) and an error is generated. You can intercept this error in the On Backup Shutdown Database Method. The error is also indicated in the Backup journal.

Backup File Destination Folder

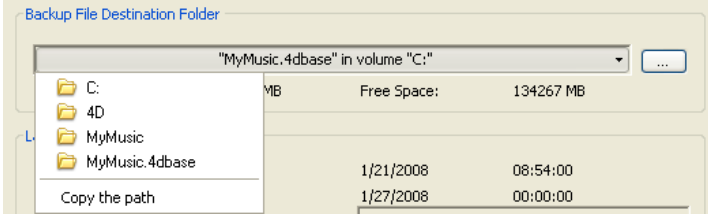
This area allows viewing and modifying the location where backup files and log files (if any) are stored.

4D generates two types of backup files: backup files and backup log files. Backup files are special for two reasons:

- They can contain several files (structure, data, attached files, etc.).
- They are secured using internal verification mechanisms (which can be set on the “Backup” page of the Preferences).

By default, 4D stores these files next to the database data file. It is strongly advised to set a location on another disk volume to reduce the risk of data loss in case of disk failure on the drive containing the database.

You can click in the area indicating the location of the backup files in order to display their pathname as a pop-up menu containing the series of folders on the disk:



If you select an item in this menu, it will be displayed in a new system window. The **Copy the path** command copies the complete pathname as text in the clipboard.

To modify the location where these files are stored, click the [...] button. A selection dialog box appears, which allows selecting a folder or disk where the backups will be placed. The "Used Space" and "Free Space" areas are updated automatically and indicate the remaining disk space on the disk of the selected folder.
You should make sure that the free space is sufficient for all of your backups. If a backup fails due to a lack of disk space, an error is generated. You can intercept this error in the On Backup Shutdown Database Method. The error is also indicated in the information area and in the Backup journal.

Backup File Names

4D names backup files using a specific naming system on which the automatic restore functions are based. This naming system cannot be changed.

Standard backups are named ***Databasename*[xxxx].4BK**, where *database* is the name of the database data file and *xxxx* is the number of the backup. For example, the 26th backup of the Invoices database is named *Invoices[0026].4BK*.
If the backup is segmented, 4D adds the segment number as *-xxxx*. For example, the 3rd segment of the 26th backup of the Invoices database is named *Invoices[0026-0003].4BK*.
For more information on segments, please refer to the "Archive" paragraph on page 1184.

- Backups of log files are named ***Logname*[xxxx].4BL**, where *logname* is the name of the log file of the database and *xxxx* is the number of the backup (starting at 0). For example, the 13th backup of the Log log file is named *Log[0012].4BL*.
If the backup of the log file is segmented, 4D adds the segment number as *-xxxx*. For example, the 2nd segment of the 13th backup of the log file Log is named *Log[0012-0002].4BL*.
For more information on segments, please refer to the "Archive" paragraph on page 1184.

Please note that log file backups start at 0 while database file backups start at 1. For the first database file backup (backup[0001].4BK for example), the log file backup is named log[0000]: it represents the changes made in the data file starting at its "empty" state and can only be integrated into an empty data file. Consequently, a log backup named, for example, log[0025].4BL must be interpreted as the "26th backup of the log file, corresponding to operations performed between the 25th and 26th database backup". log[0025].4BL thus corresponds to the backup[0025].4BK backup.

Note The backup numbering ranges are as follows:
- backup: 1 to 9999
- log file backup: 0 to 9998
- segment: 1 to 9999.

Last Backup Information The "Last Backup Information" area provides information on the last database backup. Information is provided if at least one backup has taken place.

- **Last backup**: Date and time of the last backup.
- **Next scheduled backup**: Date and time of the next backup; information is provided in this area if a backup schedule has been put into place.
- **Backup file name**: Access path and file name of the last main backup. If the backup is segmented, the name of the first segment is displayed. You can click in the area indicating the location of the file in order to display its pathname as a pop-up menu containing the series of folders on the disk.
- **Log backup file name**: Access path and file name of the last log file backup (if any). You can click in the area indicating the location of the file in order to display its pathname as a pop-up menu containing the series of folders on the disk.
- **Status**: This area displays the error code of the last backup, as well as a description of this code. If the backup was executed properly, the area indicates "No detected error."
  For scheduled backups, you can use this area to verify that the last backup occurred as scheduled.

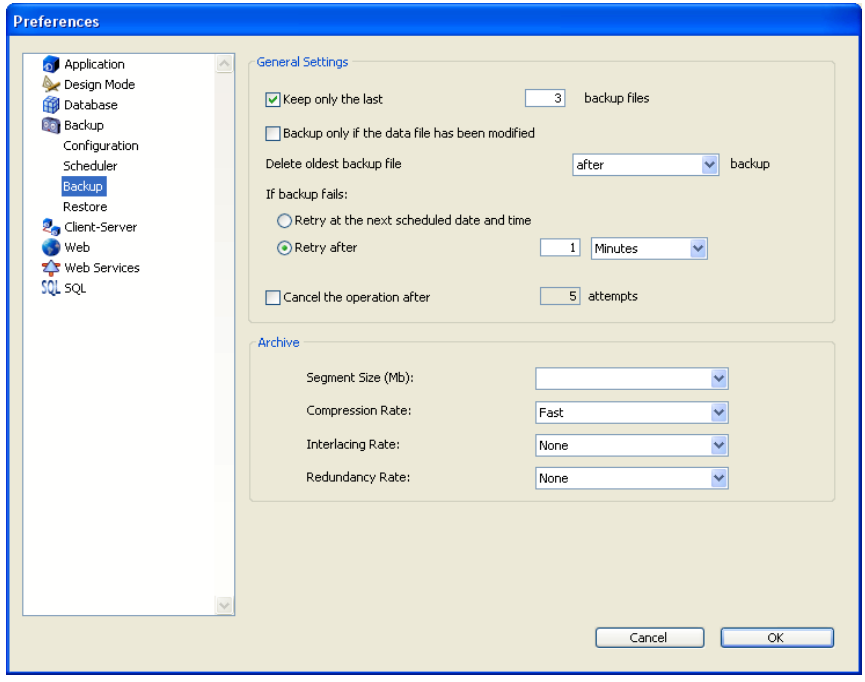
This information is also displayed on the Backup page of the MSC (see the "Backup" paragraph on page 1220).

Note The parameters related to log file management are covered in the "Managing the Log File" paragraph on page 1188.

Backup Settings

Like the configuration settings, backup settings are used for each backup. Moreover, any changes to these settings are optional. Their default values correspond to the standard use of the backup function.

The backup settings are defined on the **Backup** page of the application Preferences:



4D Server The backup settings can only be set from the 4D Server machine.

These settings are specific to each database opened with the 4D application.

Keep Only the Last X Backup Files

This parameter allows activating and configuring the mechanism used to delete the oldest backup files. It lets you keep a specific number of the last backup files on the backup disk — the oldest file is then deleted at each new backup— and thus avoids the risk of saturating the disk drive.

This feature works as follows: once the current backup is complete, 4D deletes the oldest archive if it is found in the same location as the archive to back up and has the same name (you can request that the oldest archive be deleted before the backup in order to save space). If, for example, the number of sets is set to 3, the first three backups create the archives MyBase-0001, MyBase-0002, and MyBase-0003 respectively. During the fourth backup, the archive MyBase-0004 is created and MyBase-0001 is deleted.

Based on the space on the disk that you set aside for your backups, you can determine the maximum possible number of backup sets using the following equation:

$$\text{Maximum number of sets} = \frac{\text{Space available for backups}}{\text{Maximum backup size}} - 1 \text{ set}$$

You must reduce the maximum number of sets by 1 because 4D, by default, first performs the current backup, then deletes the oldest archive from the disk. This behavior can be changed (please see the "Delete Oldest Backup File Before/After Backup" paragraph on page 1183).

By default, the mechanism for deleting sets is enabled and 4D keeps 3 backup sets.

To disable the mechanism, simply deselect the option.

Note This parameter concerns both the database backup sets and the log file backup sets.

Backup Only if the Data File has been Modified

When this option is checked, 4D starts scheduled backups only if data has been added, changed or deleted in the database since the last backup. If not, the scheduled backup is cancelled and carried over until the next scheduled backup. No error is generated; however the backup journal notes that the backup has been postponed.

This option also allows saving machine time for the backup of databases mainly used for viewing purposes. Please note that enabling this option does not take into account any modifications made to the structure file or attached files.

Notes
- For more information on scheduled backups, please refer to the "Scheduled Backup Settings" paragraph on page 1185.
- Manual backups of the database do not take this option into account.
- This parameter concerns both database and log file backups.

Delete Oldest Backup File Before/After Backup

This option is only used if the "Keep only the last X backup files" option is checked. It allows you to specify whether 4D should start by deleting the oldest archive before starting the backup (**before** option) or if the deletion should occur once the backup is completed (**after** option). In order for this mechanism to work, the oldest archive must not have been renamed or moved.

The **before** option allows you to save space. It is not necessary to reduce the number of backups allowed by 1, which lets you, for example, store two 2-GB archives on a 5-GB disk. Please also note that in cases of interruption during a backup, you can access neither the old backup (since it was previously deleted), nor the current backup, which is not yet completed. Since the database could not be modified during the backup, you can be sure that your data file is intact, even if you do not have a valid backup. Once the system restarts and the incomplete archive is destroyed, proceed immediately with a new backup.

The **after** option is an additional security measure but requires more space on the backup hard disk.

If Backup Fails

This option allows defining the mechanism used to handle failed backups (backup impossible).

Notes
- Not all incidents result in a failed backup (please refer to the "If Backup Fails" paragraph on page 1183).
- 4D considers a backup as failed if the database was not launched at the time when the scheduled automatic backup was set to be carried out (please refer to the "Scheduled Backup Settings" paragraph on page 1185).

When a backup cannot be performed, 4D allows making a new attempt. Two options are available:

- **Retry at the next scheduled date and time**: This option only makes sense when working with scheduled automatic backups (please refer to the "Scheduled Backup Settings" paragraph on page 1185). It cancels the failed backup and an error is generated.
- **Retry after X Seconds, Minutes or Hours**: When this option is checked, a new backup attempt is executed after the wait period. This mechanism allows anticipating certain circumstances that block the backup. You can set a wait period in seconds, minutes or hours using the corresponding menu. If the new attempt also fails, an error is generated and the failure is noted in the status area of the last backup and in the backup journal file.
- **Cancel the operation after X attempts**: This parameter is used to set the maximum number of failed backup attempts.
  If the backup has not be carried out successfully after the maximum number of attempts set has been reached, it is cancelled and the error 1401 is generated ("The maximum number of backup attempts has been reached; automatic backup is temporarily disabled"). In this case, no new automatic backup will be attempted as long as the application has not been restarted, or a manual backup has not been carried out successfully.
  This parameter is useful in order to avoid a case where an extended problem (requiring human intervention) that prevented a backup from being carried out would have led to the application repeatedly attempting the backup to the detriment of its overall performance.
  By default, this parameter is not checked.

Archive

This area allows setting archive generation options. These options apply to main backup files and to log file backup files.

- **Segment Size (MB):**
  4D allows you to segment archives, which is to cut an archive up into smaller sizes. This behavior allows you, for example, to store a backup on several different disks (CDs, ZIPs, etc.). During a restore process, 4D will automatically fuse the segments. Each segment is called *MyDatabase[xxxx-yyyy].4BK*, where *xxxx* is the backup number and *yyyy* is the segment number. For example, the three segments of the sixth MyDatabase database backup will be called MyDatabase[0006-0001].4BK, MyDatabase[0006-0002].4BK and MyDatabase[0006-0003].4BK.

The **Segment Size** menu is a combo box that allows you to set the size in MB for each segment of the backup. You can choose one of the preset sizes or enter a specific size between 0 and 2048. If you pass 0, no segmentation occurs (this is the equivalent of passing **None**).

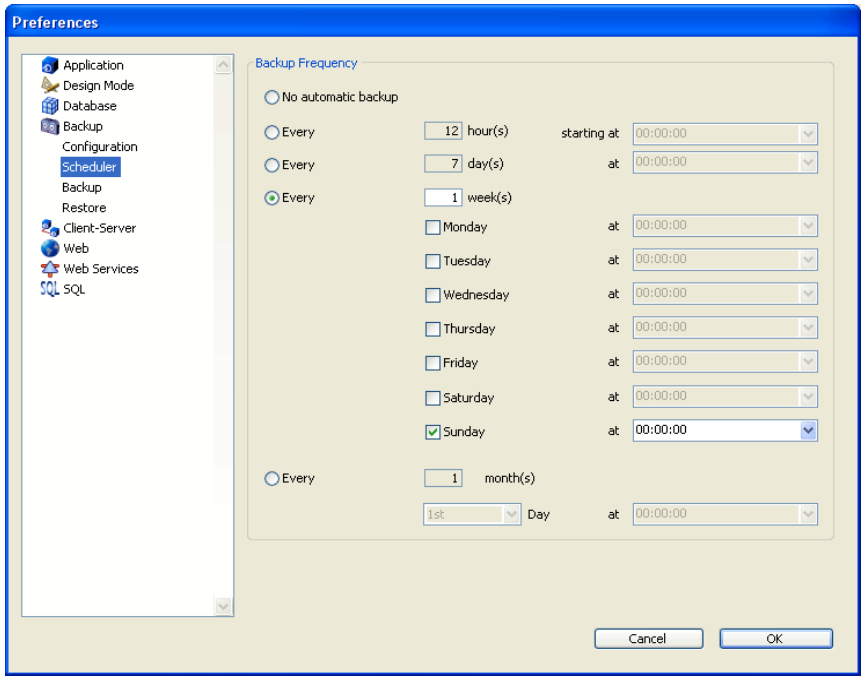
- **Compression Rate**
  By default, 4D compresses backups to help save disk space. However, the file compression phase can noticeably slow down backups when dealing with large volumes of data.
  The **Compression Rate** option allows you to adjust file compression:
  - **None**: No file compression is applied. The backup is faster but the archive files are considerably larger.
  - **Fast** (default): This option is a compromise between backup speed and archive size.
  - **Compact**: The maximum compression rate is applied to archives. The archive files take up the least amount of space possible on the disk, but the backup is noticeable slowed.
- **Interlacing Rate** and **Redundancy Rate**
  4D generates archives using specific algorithms that are based on optimization (interlacing) and security (redundancy) mechanisms. You can set these mechanisms according to your needs. For these two options, 4D provides a default rate **Medium**. The menus for these options also contain rates of **Low**, **High** and **None**.
  - **Interlacing Rate**: Interlacing consists of storing data in non-adjacent sectors in order to speed up their read time. However, the storage phase is slower.
  - **Redundancy Rate**: Redundancy allows securing data present in a file by repeating the same information several times. The higher the redundancy rate, the better the file security; however, storage is slow and the file size is large.

Scheduled Backup Settings

You can automate the backup of databases opened with 4D or 4D Server (even when no client machines are connected). This involves setting a backup frequency (in hours, days, weeks or months); for each session, 4D automatically starts a backup using the backup settings defined in the Preferences.

If this application was not launched at the theoretical moment of the backup, the next time 4D is launched, it considers the backup as having failed and applies the appropriate configuration, set in the database Preferences (refer to the "If Backup Fails" paragraph on page 1183).

The scheduler backup settings are defined on the **Scheduler** page of the application Preferences:



First select a frequency scale (days, weeks, etc.) by clicking on the corresponding radio button. You must then specify when to perform the backup.

- **No automatic backup**: The scheduled backup feature is disabled.
- **Every X hour(s)**: This option allows programming backups on an hourly basis. You can enter a value between 1 and 24.
  - **starting at**: Sets the time at which the first hourly backup will begin.
- **Every X day(s) at x**: This option allows programming backups on a daily basis. Enter 1 if you want to perform a daily backup. When this option is checked, you must enter the time when the backup should start.

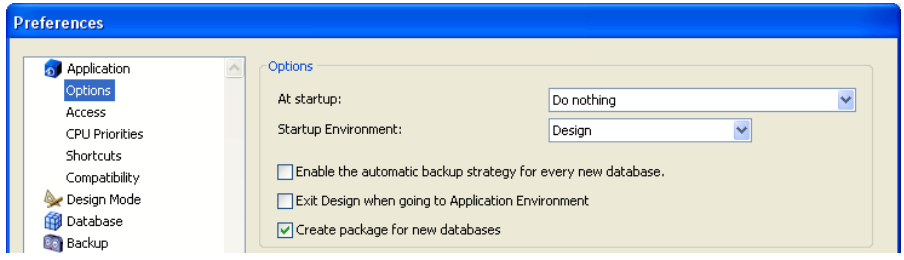
- **Every X week(s) day at x**: This option allows programming backups on a weekly basis. Enter 1 if you want to perform a weekly backup. When this option is checked, you must enter the day(s) of the week and the time when the backup should start. You can select several days of the week, if desired. For example, you can use this option to set two weekly backups: one on Wednesday and one on Friday.
- **Every X month(s), Xth Day at x**: This option allows programming backups on a monthly basis. Enter 1 if you want to perform a monthly backup. When this option is checked, you must indicate which day of the month and the time when the backup should start.

Automatic Backup Strategy

4D provides a default backup strategy when a database is created. This strategy puts into place a set of backup parameters that assure minimum database security.

These settings can be changed once the database is created.

The backup strategy is enabled, by default, when a new database is created. To disable it, simply deselect the **Enable the automatic backup strategy for every new database** option in the Preferences dialog box (on the Application/Option page):



The automatic backup strategy includes the following settings:

- Creation and use of a log file (named *DataFileName.journal*).
- Backup of all database files (data, log, structure and user structure) next to the database structure file.
- Automatic backup every Sunday at 00:00:00.
- Default backup settings (keeping last 3 backups, medium rates, etc.).
- All the automatic restore options.

Of course, it is also possible to change these settings after the database has been created.

Managing the Log File

A continuously-used database is always recording changes, and record additions or deletions. Performing regular backups of data is important but does not allow (in case of incident) restoring data entered since the last backup. To respond to this need, 4D now offers a specific tool: the log file. This file allows ensuring permanent security of database data.

In addition, 4D works continuously with a data cache in memory. Any changes made to the data of the database are stored temporarily in the cache before being written to the hard disk. This accelerates the operation of applications; in fact, accessing memory is faster than accessing the hard disk. If an incident occurs in the database before the data stored in the cache could be written to the disk, you must include the current log file in order to restore the database entirely.

Introduction

The log file generated by 4D contains all operations performed on a database, which are logged sequentially. As such, each operation performed by a user causes two simultaneous actions: the first one in the database (instruction is executed normally) and the second one in the log file (the description of the operation is recorded). The log file is created independently without disturbing or slowing down the work of the user. A database can only work with one log file at a time.

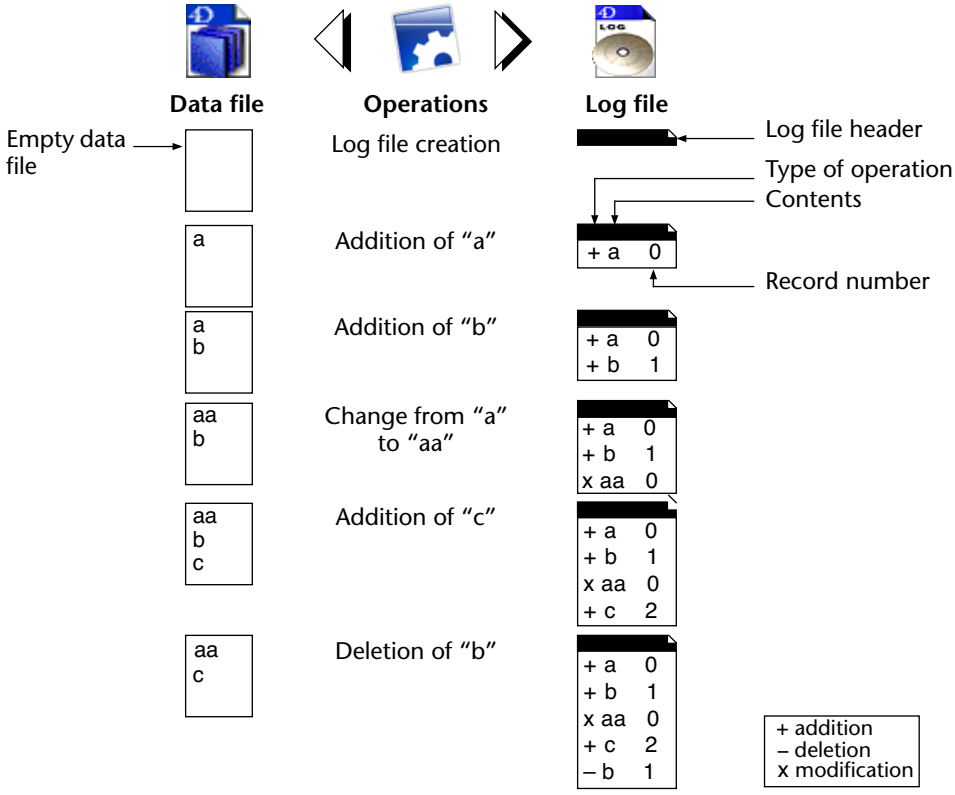
The log file records the following types of operations:

- Opening and closing of the data file,
- Opening and closing of the process (contexts),
- Adding of records or BLOBs
- Modifying of records
- Deleting of records
- Creating and closing of transactions.

Note For more information about these actions, please refer to the "Parsing a Log File" paragraph on page 1193.

4D handles the log file. It takes into account all operations that affect the data file equally, regardless of any manipulations performed by a user, 4D methods, the SQL engine, 4D plug-ins (4D Write, 4D View, etc.), a Web browser or even 4D Open when using 4D Server.

The following illustration sums up how the log file works:



The current log file is automatically saved with the current data file. This mechanism has two distinct advantages:

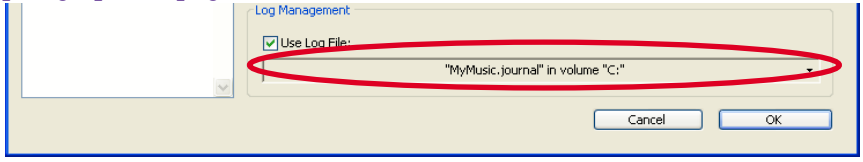
- Its avoids saturating the disk volume where the log file is stored. Without a backup, the log file would get bigger and bigger with use, and would eventually use all available disk space. For each data file backup, 4D or 4D Server closes the current log file and immediately starts a new, empty file, thereby avoiding the risk of saturation. The old log file is then archived and eventually destroyed depending on the mechanism for managing the backup sets.

- It keeps log files corresponding to backups in order to be able to parse or repair a database at a later point in time. The integration of a log file can only be done in the database to which it corresponds. It is important, in order to be able to properly integrate a log file into a backup, to have backups and log files archived simultaneously.

Creating the Log File

By default, any database created with 4D uses a log file: the creation of this file is part of the automatic backup strategy (please refer to the "Automatic Backup Strategy" paragraph on page 1187).
The log file is named **DataFileName.journal** and is placed in the folder containing the database structure.

You can find out if your database uses a log file at any time: just verify whether the **Use Log File** option is checked on the **Configuration** page of the Preferences (please refer to the "Configuration of Backup Files" paragraph on page 1176).

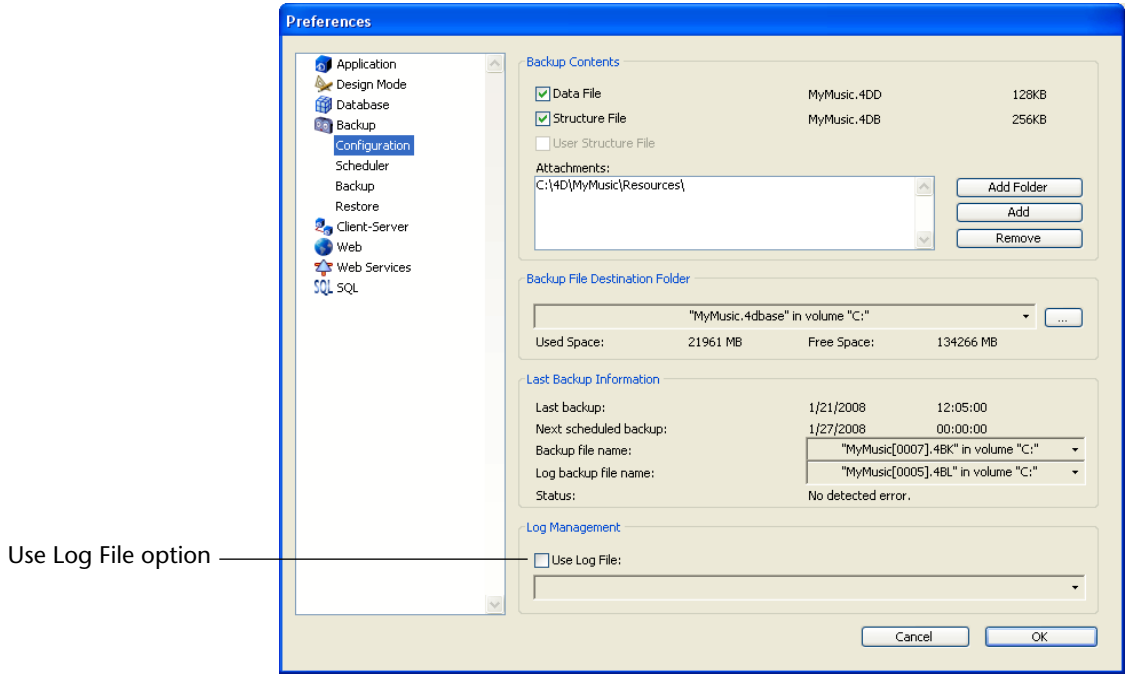


With 4D Server, the name of the log file is also listed in the server machine window.

If you deselected this option, or if you use a database without a log file and wish to set up a backup strategy with a log file, you will have to create one.

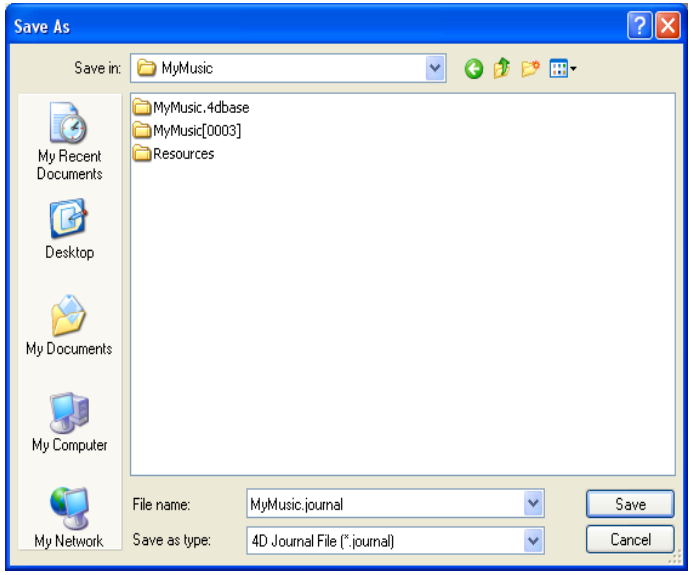
► To create a log file:

1 On the Configuration page ("Backup" theme) of the application Preferences, check the Use Log File option.



Use Log File option

The program displays a standard open file or new log file dialog box:



By default, the log file is named **DataFileName.journal**.

2 **Keep the default name or rename it, and then select the file location.**

If you have at least two hard drives, it is recommended that you place the log file on a disk other than the one containing the database. If the database hard drive is lost, you can still recall your log file.

3 **Click Save.**

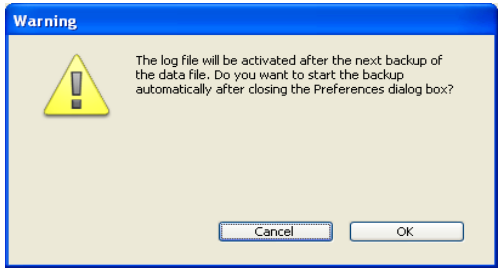
The disk and the name of the open log file are now displayed in the "Use Log File" area of the Preferences dialog box. You can click on this area in order to display a pop-up menu containing the series of folders on the disk.

4 **Validate the Preferences dialog box.**

In order for you to be able to create a log file directly, the database must be in one of the following situations:

- The data file is blank,
- You just performed a backup of the database and no changes have yet been made to the data.

In all other cases, when you validate the Preferences dialog box, an alert dialog box will appear to inform you that it is necessary to perform a backup.



If you click **OK**, the backup begins immediately, then the log file is activated. If you click **Cancel**, the request is saved but the creation of the log file is postponed and it will actually be created only after the next backup of the database.

This precaution is indispensable because, in order to restore a database after any incidents, you will need a copy of the database into which the operations recorded in the log file will be integrated.

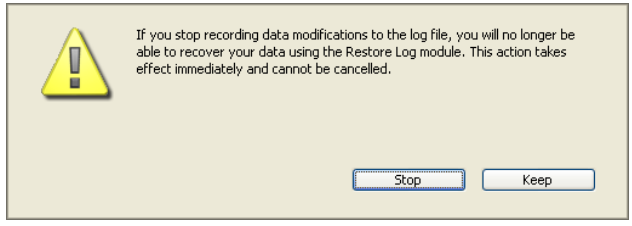
Without having to do anything else, all operations performed on the data are logged in this file and it will be used in the future when the database is opened.

- You must create another log file if you create a new data file.
- You must set or create another log file if you open another data file that is not linked to a log file (or if the log file is missing).

Stopping a Log File

If you would like to stop logging operations to the current log file, simply deselect the **Use Log File** option on the **Configuration** page ("Backup" theme) of the application Preferences.

4D then displays an alert message to remind you that this action prevents you from taking advantage of the security that the log file provides:



If you click **Stop**, the current log file is immediate closed (the Preferences dialog box does not need to be validated afterwards).

If you wish to close the current log file because it is too large, you must perform a data file backup, which will cause the log file to be backed up as well. Note that there is an automatic mechanism that prevents the log file from becoming too large (see the following paragraph).

Automatic Backup of Log File When Critical Size is Reached

When the log file approaches the critical size of 2 GB, 4D automatically triggers a backup of the database, in order to close the current log file and open a new one. This avoids a situation where the size of this file could alter application performance.

In this case, the backup carried out uses the current backup parameters and is the same as a manual backup.

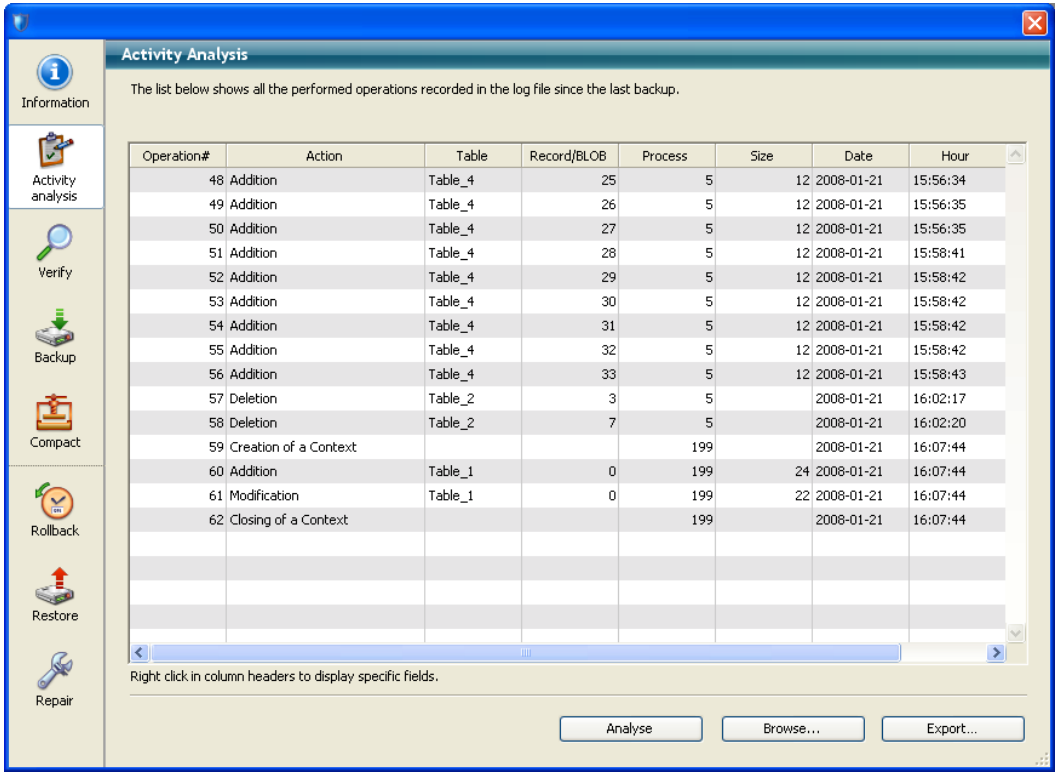
Parsing a Log File

4D includes a function that allows viewing the contents of the current log file. This function is useful for parsing the use of a database or detecting the operation(s) that caused errors or malfunctions. In the case of a database in client-server mode, it allows verifying operations performed by each client machine.

Note It is also possible to rollback the operations carried out on the data of the database. For more information about this point, please refer to the "Undoing Operations" paragraph on page 1204.

The log file analysis and rollback functions are available in the Maintenance and Security Center (MSC).

To view the current log file, select **Maintenance Security Center** from the **Help** menu of 4D (or click on the MSC icon in the 4D tool bar) and display the "Activity analysis" page:



Notes
- Only the Administrator and Designer of the database can access the log file verification function.
- For more information about the MSC, please refer to the chapter "Maintenance and Security Center" on page 1209.

Every operation recorded in the log file appears as a row. The columns provide various information on the operation. You can modify the default column display — for example, you can display field values — using the contextual menu of the window (see the "Setting the Column Display" paragraph on page 1196). Moreover, the date and time when the data file was opened and/or closed is also displayed.

This information allows you to identify the source and context of each operation:

- **Operation#**: Sequence number of operation in the log file.
- **Action**: Type of operation performed on the data. This column can contain one of the following operations:
  - *Opening of Data File*: Opening of a data file.
  - *Closing of Data File*: Closing of an open data file.
  - *Creation of a Context*: Creation of a process that specifies an execution context.
  - *Closing of a Context*: Closing of process.
  - *Addition*: Creation and storage of a record.
  - *Adding a BLOB*: Storage of a BLOB in a BLOB field.
  - *Deletion*: Deletion of a record.
  - *Modification*: Modification of a record.
  - *Start of Transaction*: Transaction started.
  - *Validation of Transaction*: Transaction validated.
  - *Cancellation of Transaction*: Transaction cancelled.
- **Table**: Table to which the added/deleted/modified record or BLOB belongs.
- **Record/BLOB**: Record number or sequence number of the BLOB involved in the operation.
- **Process**: Internal number of process in which the operation was carried out. This internal number corresponds to the context of the operation.
- **Size**: Size (in bytes) of data processed by the operation.
- **Date** and **Hour**: Date and hour when the operation was performed.
- **User**: Name of the user that performed the operation. In client-server mode, the name of the client-side machine is displayed; in single-user mode, the ID of the user is displayed.

If the 4D passwords are not enabled, this column is blank.
If the operation was performed using 4D Open, the "4D Open" label is displayed.

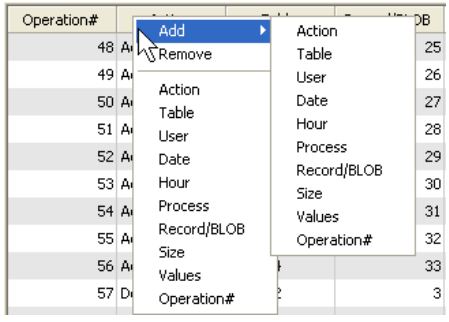
- **Values** (column not displayed by default): Values of fields for the record in the case of addition or modification. The values are separated by ";". Only values represented in alphanumeric form are displayed.

Click on **Analyze** to update the contents of the current log file of the selected database (named by default *dataname.journal*).
The **Browse...** button can be used to select and open another log file for the database.
The **Export...** button can be used to export the contents of the file as text.

Setting the Column Display

You can customize the column display using the contextual menu of the window. You can add or remove columns, or replace the contents of a column for the current window session (the customized window contents are reset when the session is closed).

To display the contextual menu of the window, click on a column header:



The following commands are available:

- **Add**: allows adding a column to the right of the column clicked on.
- **Remove**: allows removing the column clicked on.
- ***Column Names***: allows replacing the contents of the column clicked on by another column of information.

Restoring Data

The built-in 4D backup module allows you to restore entire sets of database data in case of any incidents, regardless of the cause of the incident.

Incidents and Diagnostic

Two primary categories of incidents can occur:

- **The unplanned stoppage of a database while in use.**
  This incident can occur because of a power outage, system element failure, etc. In this case, depending on the current state of the data cache at the moment of the incident, the restore of the database can require different operations:
  - If the cache was empty, the database opens normally. Any changes made in the database were recorded. This case does not require any particular operation.
  - If the cache contains operations, the data file is intact but it requires integrating the current log file.
  - If the cache was in the process of being written, the data file is probably damaged. The last backup must be restored and the current log file must be integrated.
- **The loss of database file(s).**
  This incident can occur because of defective sectors on the disk containing the database, a virus, manipulation error, etc.
  The last backup must be restored and then the current log file must be integrated.

  To find out if a database was damaged following an incident, simply relaunch the database using 4D. The program performs a self-check and details the necessary restore operations to perform.
  In automatic mode, these operations are performed directly without any intervention on the part of the user (please refer to the following paragraph).

  If a regular backup strategy was put into place, the 4D restore tools will allow you to recover (in most cases) the database in the exact state it was in before the incident. Two main tools are at your disposal:

- Backup restore — which consists in extracting files contained in one or more backup file(s) and making them readable again for the application that created them.

- Log file integration.

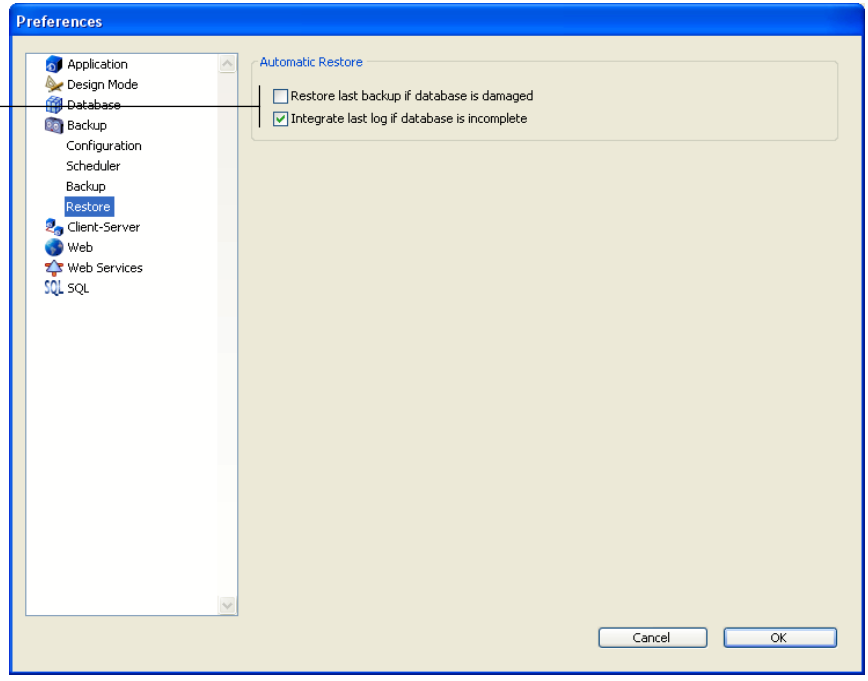
These two functions can be combined and automated.

Automatic Restore

By default, 4D automatically launches the database restore procedures after an incident. Two types of automatic features are used:

- Automatic restore
- Automatic log file integration

These mechanisms can be disabled using the options available on the **Restore** page ("Backup" theme) in the application Preferences:



Restore Last Backup if Database is Damaged

When this option is checked, the program automatically starts the restore of the data file of the last valid backup of the database, if an anomaly is detected during database launch. No intervention is required on the part of the user; the operation is logged in the backup journal (please refer to the "Backup Journal" paragraph on page 1206).

Note In the case of an automatic restore, only the data file is restored. If you wish to recover the attached files or the structure file, you must perform a manual restore.

The steps of the automatic restore are as follows:

- First, 4D renames the damaged data file.
- 4D extracts the data file of the last backup and stores it in place of the previous one.
- If the "Integrate last log if database is incomplete" option is checked, 4D integrates the log file, if necessary (please refer to the "Integrate Last Log File if Database is Incomplete" paragraph on page 1199).
- 4D opens the restored database.

Integrate Last Log File if Database is Incomplete

When this option is checked, the program automatically integrates the log file when a restored database is opened. No intervention on the part of the user is required.

- When opening a database, the current log file is automatically integrated if 4D detects that the operations stored in the log file are not present in the data. This situation occurs, for example, when a power outage occurs when there are operations in the data cache that have not yet been written to the log.
- When restoring a database, if the current log file or backup log file having the same number as the backup file is stored in the same folder, 4D examines its contents. If it contains operations not found in the data file, the program automatically integrates it.

  The user does not see any dialog box; the operation is completely automatic. The goal is to make use as easy as possible. The operation is logged in the backup journal (please refer to the "Backup Journal" paragraph on page 1206).

Manually Restoring a Backup (Standard Dialog)

You can restore the contents of an archive generated by the backup module manually. A manual restore may be necessary, for instance, in order to reproduce the contents of an archive in full (structure files and/or enclosed attached files), or for the purpose of carrying out searches among the archives.

The manual restore can also be performed along with the integration of the current log file.

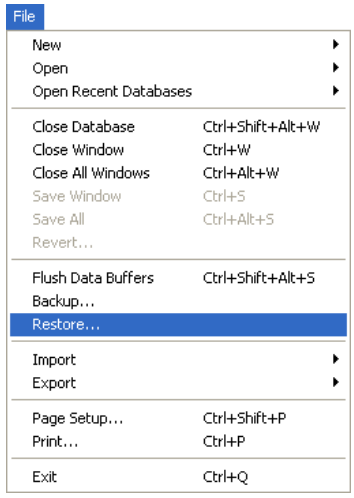
Note You can set 4D so that the database data are automatically restored in case of any incidents. For more information on this, please refer to the "Automatic Restore" paragraph on page 1198.

The manual restore of backups can be carried out either via the standard Open document dialog box, or via the “Restore” page of the Maintenance and Security Center (MSC).

- Restoring via a standard dialog box can be used to restore any archive. This function is described below.
- Restoring via the MSC provides more options and allows the archive contents to be previewed. On the other hand, only archives associated with the open database can be restored. This function is described in the “Manually Restoring a Backup (MSC)” paragraph on page 1202.

► To restore a database manually via a standard dialog box:

1 Start the 4D application and choose Restore... in the File menu.



It is not mandatory that a database be open.

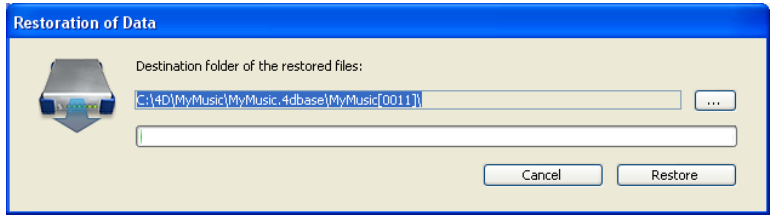
OR

Execute the RESTORE command from a 4D method.

A standard Open file dialog box appears so that you can indicate the backup file (.4bk) or the backup log file (.4bl) to be restored.

2 Select the file to restore and click Open.

The following dialog box appears, which allows you to specify the location where files will be restored:



By default, 4D restores the files in a folder named "*Archivename*" (no extension) located next to the archive. You can click on the **[...]** button to specify a different location.

3 Click on the Restore button.

4D extracts all backup files from the specified location.

If the current log file or a backup log file with the same number as the backup file is stored in the same folder, 4D examines its contents. If it contains operations not present in the data file, the program asks you if you want to integrate these operations.

Note Integration is done automatically if the "Integrate last log file..." option is checked (please refer to the "Integrate Last Log File if Database is Incomplete" paragraph on page 1199).

4 (Optional) Click OK to integrate the log file into the restored database.

If the restore and integration were carried out correctly, 4D displays a dialog box indicating that the operation was successful.

5 Click OK .

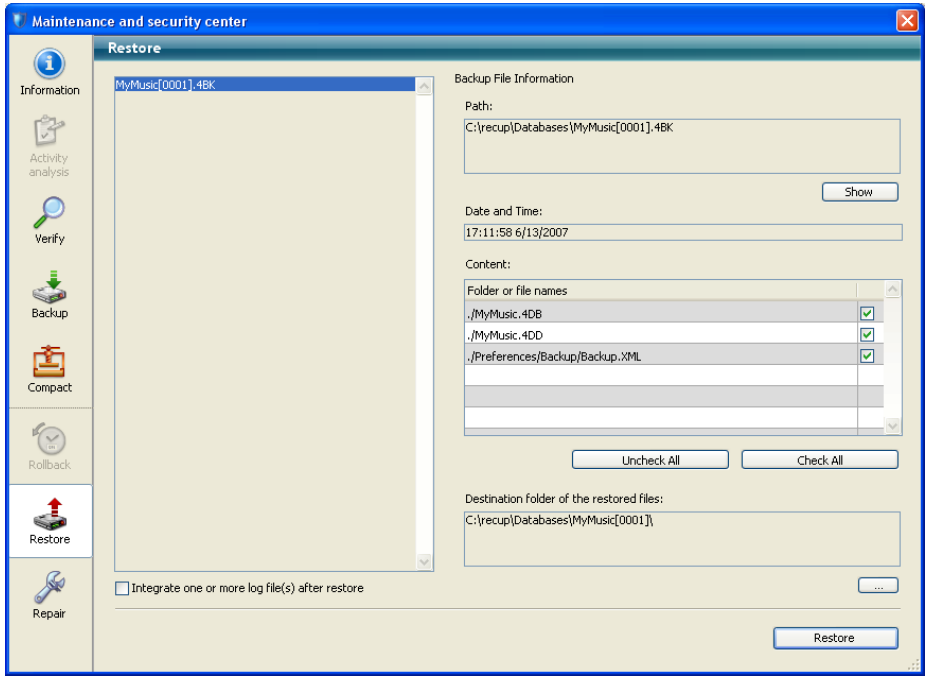
The destination folder is displayed. During the restore, 4D places all backup files in this folder, regardless of the position of the original files on the disk when the backup starts. This way your files will be easier to find.

Manually Restoring a Backup (MSC)

You can manually restore an archive of the current database using the "Restore" page of the Maintenance and Security Center (MSC). This page provides several options that can be used to control the restore.

Note It is also possible to manually restore any archive via a standard Open document dialog ox (see the "Manually Restoring a Backup (Standard Dialog)" paragraph on page 1199).

To restore a backup of the current database, select **Maintenance Security Center** in the **Help** menu of 4D (or click on the MSC icon in the 4D tool bar) and display the "Restore" page:



Note For more information about the MSC, please refer to the chapter "Maintenance and Security Center" on page 1209.

The list found in the left part of the window displays any existing backups of the database. When you select a backup in this list, the right part of the window displays the information concerning this particular backup:

- **Path**: Complete pathname of the selected backup file. Clicking the **Show** button opens the backup file in a system window.

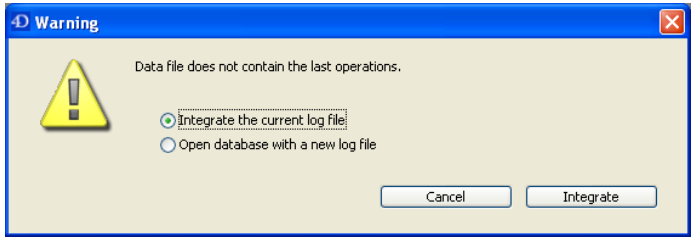
- **Date and Time**: Date and time of backup.
- **Content**: Contents of the backup file. Each item in the list has a check box next to it which can be used to indicate whether or not you want to restore it. You can also use the **Check All** or **Uncheck All** buttons to set the list of items to be restored.
- **Destination folder of the restored files**: Folder where the restored files will be placed. By default, 4D restores the files in a folder named "*Archivename*" (no extension) that is placed next to the database structure file. To change this location, click on [...] and specify the folder where you want the restored files to be placed.

Successive Integration of Several Log Files

The **Integrate one or more log file(s) after restore** option allows you to integrate several log files successively into a database. If, for example, you have 4 log archives (corresponding to 4 database backups), you can restore the first backup then integrate the log archives one by one. This means that you can, for example, recover a data file even when the last backup files are missing.
When this option is checked, 4D displays the standard Open file dialog box after the restore, which can be used to select log file to be integrated. The Open file dialog box is displayed again after each integration until it is cancelled.

Manually Integrating the Log

If you have not checked the option for the automatic integration of the log file (see the "Integrate Last Log File if Database is Incomplete" paragraph on page 1199), a warning dialog box appears during the opening of the database when 4D notices that the log file contains more operations than have been carried out in the database.



In order for this mechanism to work, 4D must be able to access the log file in its current location.

You can choose whether or not to integrate the current log file. Not integrating the current log file allows you to avoid reproducing errors made in the data.

Undoing Operations

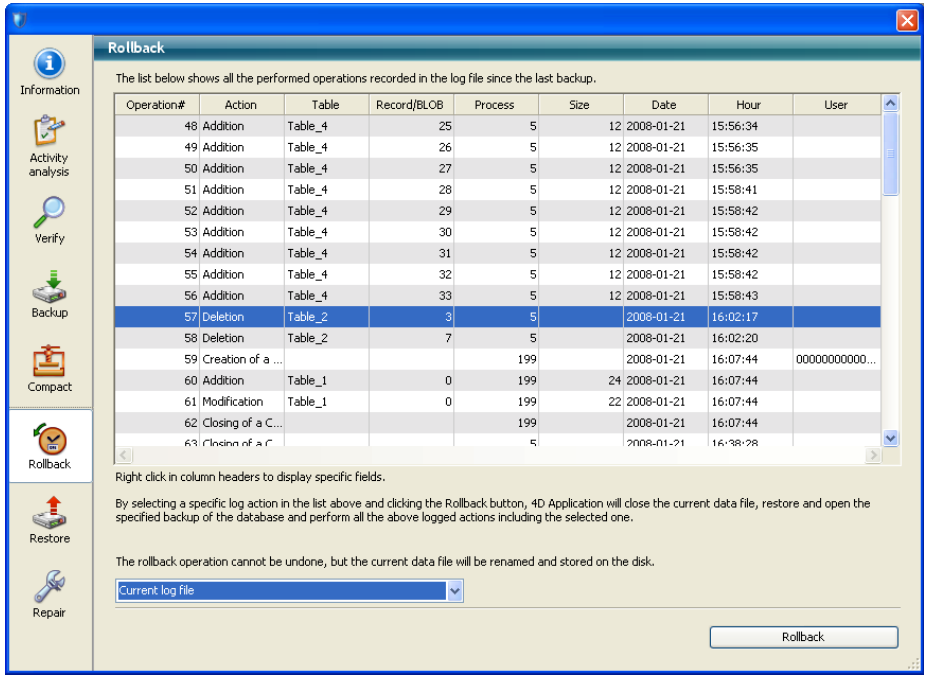
4D features a function that allows undoing operations performed on the log file. This function is similar to a multi-level cancel. It is especially useful when a record was deleted by error from a database.

In order for this function to be available, the database must have a log file.

► To restore a database to a previous state:

1 Select Maintenance Security Center in the Help menu of 4D and display the "Rollback" page.

The "Rollback" window appears:

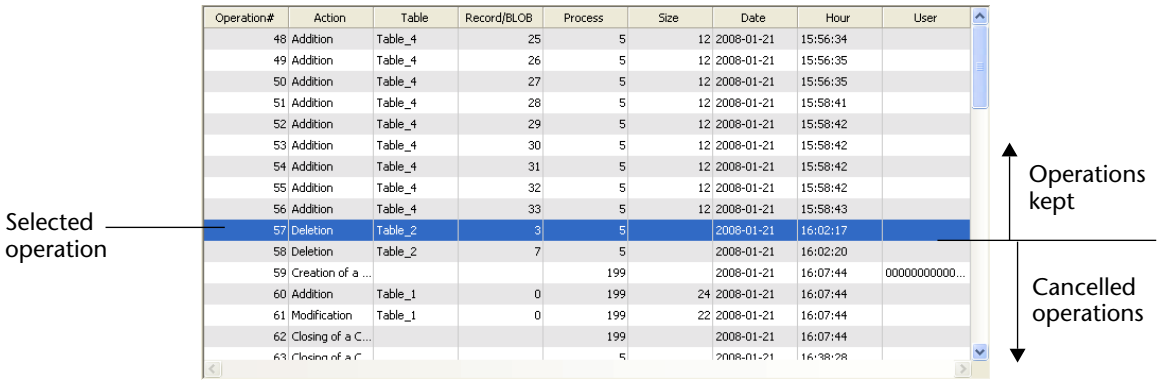


Note Only the Administrator and Designer of the database have access to check the log file.

The list of operations is identical to that of the Activity analysis window. For more information about this list, please refer to the "Parsing a Log File" paragraph on page 1193).

2 Select the row after which all operations must be cancelled.

The operation of the selected row will be the last kept. If, for example, you wish to cancel a deletion, select the operation located just before it. The deletion operation, as well as all subsequent operations, will be cancelled:



3 Click Roll back.

4D asks you to confirm the operation.

4 Click OK.

The data is then restored to the exact state it was in at the moment of the selected action.

The menu found at the bottom of the window can be used to select a log file to be used when you apply the rollback function to a restored database. In this case, you must specify the log file corresponding to the archive.

How it Works

When the user clicks the **Roll back** button, 4D shuts the current database and restores the last backup of the database data. The restored database is then opened and 4D integrates the operations of the log file up through to the selected operation.
If the database has not yet been saved, 4D starts with a blank data file.

Backup Journal

To make following up and verifying database backups easier, the backup module writes a summary of each operation performed in a special file, which is similar to an activity journal. Like an on-board manual, all database operations (backups, restores, log file integrations) are logged in this file whether they were scheduled or performed manually. The date and time that these operations occurred are also noted in the journal.

The backup journal is named “Backup Journal.txt” and is placed next to the database data file.

The backup journal can be opened with any text editor. To make parsing easier, information in the journal is separated with tabs and each line of information ends with a carriage return.

Management of Backup Journal Size

In certain backup strategies (for example, in the case where numerous attached files are being backed up), the backup journal can quickly grow to a large size.
Two mechanisms can be used to control this size:

- Automatic backup,
- Possibility of reducing the amount of information recorded.

Automatic Backup

An automatic mechanism can be used to limit the size of the backup journal: before each backup, the application examines the size of the current backup journal file. If it is greater than 10 MB, the current file is archived and a new file is created. The archived files are renamed “Backup Journal[xxx].txt”, where *xxx* is a number from 1 to 999. Once file number 999 is reached, the numbering begins at 1 again and the existing files will be replaced.

Reduce Amount of Information Stored

It is possible to reduce the amount of information recorded in the backup journal. To do this, simply modify the value of the VerboseMode key in the *Backup.xml* file of the database. By default, this key is set to True.
If you change the value of this key to False, only the main information will be stored in the backup journal: date and time of start of operation and any errors encountered.

The XML keys concerning backup configuration are described in the *XML Keys-Backup* manual.

Using a backup.xml File

The backup preferences of 4D are available as an XML file associated with the database. 4D uses the data of this file to set the backup options (options found in the Preferences dialog box) when each backup is launched. This file can also be used to read or set additional options, such as the amount of information stored in the backup journal (see previous paragraph).

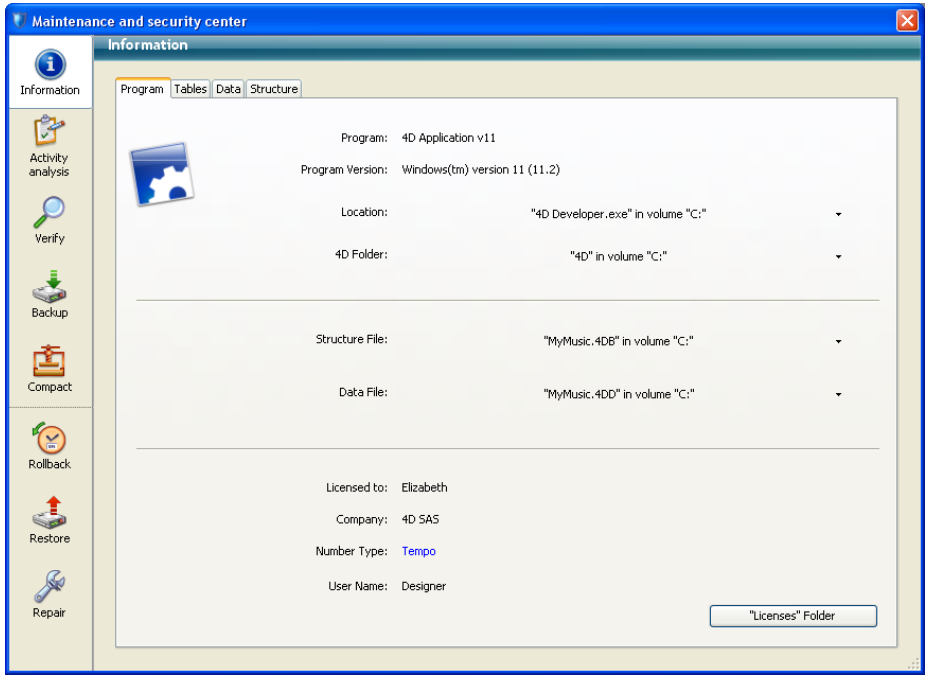
The XML backup configuration file is named **backup.xml**. It is automatically created at the following location:
DatabaseFolder/Preferences/Backup/Backup.xml

... where *DatabaseFolder* is the folder containing the database structure file. The Preferences/Backup subfolders are automatically created when necessary.

The XML keys concerning backup configuration are described in the *XML Keys-Backup* manual.

25 Maintenance and Security Center

The Maintenance and Security Center (MSC) window groups together a full set of tools to check, analyze, maintain, back up and compact both data and structure files:



The navigation control panel at the left edge of the window lets you select the theme of operation or information to display.

The MSC is available from all 4D applications: 4D Developer, 4D Server or 4D Desktop.

Displaying the MSC

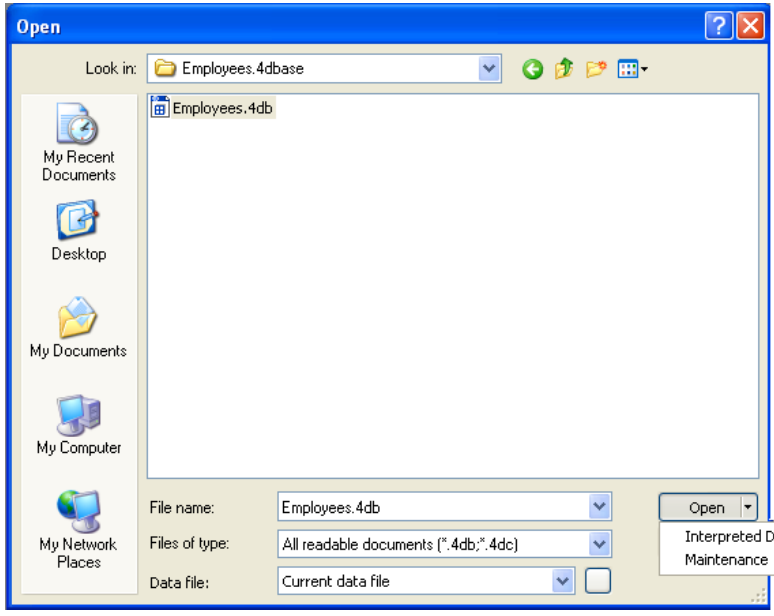
There are several ways to open the MSC window. The way it is accessed also determines the way the database is opened: in “maintenance” mode or “standard” mode. In maintenance mode, the database is not opened by 4D, only its reference is provided to the MSC. In standard mode, the database is opened by 4D.

Display in Maintenance Mode

In maintenance mode, only the MSC window is displayed (the database is not opened by the 4D application). This means that databases that are too damaged to be opened in standard mode by 4D can nevertheless be accessed. Moreover, certain operations (compacting, repair, and so on) require the database to be opened in maintenance mode (see the paragraph “Access Rights” on page 1211).

You can open the MSC in maintenance mode from two locations:

- **From the standard opening dialog box**
  The standard Open database dialog box includes the **Maintenance & Security Center** option from the menu associated with the **Open** button:



You can then simply designate the database to examine and click on **Open**.

- **Help/Maintenance Security Center menu or MSC button in the tool bar (database not open)**
  When you call this function, a standard Open file dialog box appears so that you can indicate the database to be examined. The database will not be opened by 4D.

Display in Standard Mode

In standard mode, a database is open. In this mode, certain maintenance functions are not available. You have several possibilities:

- **Help/Maintenance Security Center menu or MSC button in the tool bar** in the Design mode.
  This command opens the MSC window. This feature is not available in Application mode.
- **Via the "Open MSC" standard action** that it is possible to associated with a created menu command.
- **Using the OPEN SECURITY CENTER language command**.

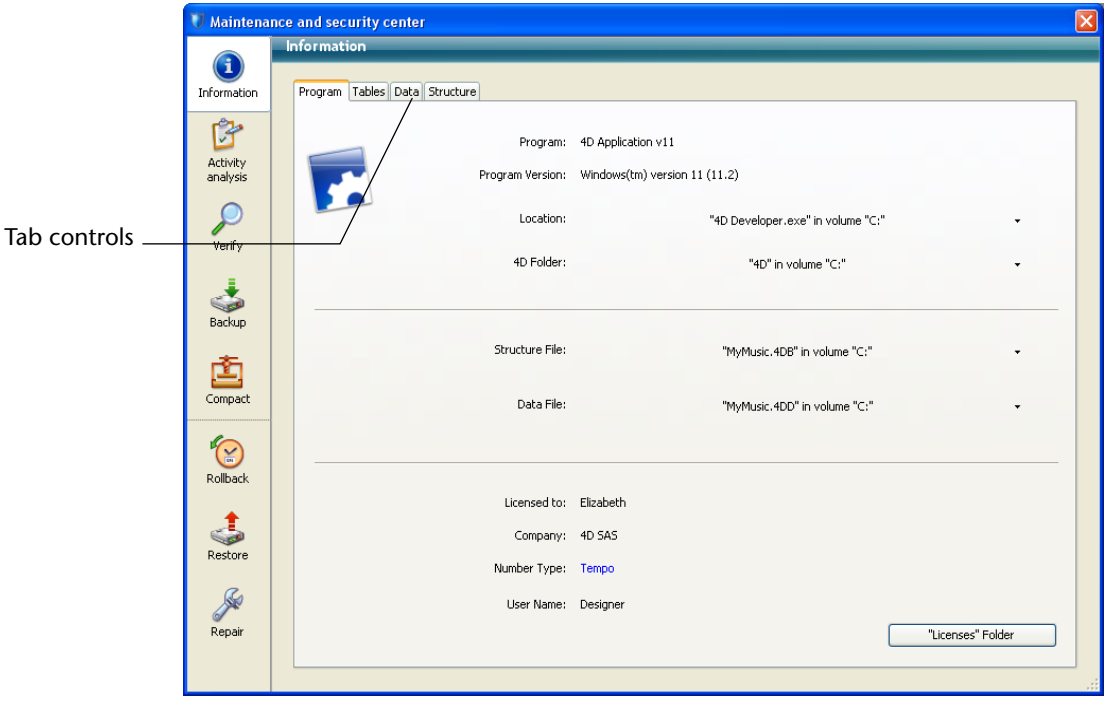
Access Rights

Certain MSC functions are not available depending on the type of application, the MSC opening mode or (when passwords are activated) the user profile:

- Functions which affect the application structure (checking, repair and compacting) can only be accessed from 4D Developer and 4D Server applications.
  In the 4D Client and 4D Desktop applications, the corresponding buttons and tabs are hidden.
- Information concerning the contents of data and structure files are only available when the database is open (the MSC must have been opened in standard mode).
- Data compacting, rollback, restore and repair functions can only be used with data files that are not open (the MSC must have been opened in maintenance mode). If these functions are solicited while the database is open in standard mode, a dialog box is displayed to let you restart the application in maintenance mode.
- If passwords have been activated, data compacting, rollback, restore and repair functions can only be accessed by the Administrator and Designer.

Information

The “Information” page provides information about the 4D environment and the system environment, the database and the application files. Each page can be displayed using tab controls at the top of the window:



Program and Tables

The **Program** and **Tables** pages provide environment information concerning the 4D application and the database structure.

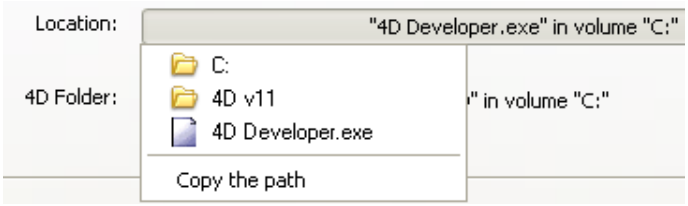
Program

This page indicates the name, version and location of the application as well as the active 4D folder (for more information about the active 4D folder, please refer to the description of the Get 4D folder command in the 4D *Language Reference* manual).

The central part of the window indicates the name and location of the database structure and data files.

The lower part of the window indicates the name of the 4D license holder and the type of license, as well as the name of the database user when passwords have been activated (or Designer if this is not the case).

- **Display and selection of pathnames**: On the **Program** page, pathnames are displayed in pop-up menus containing the folder sequence as found on the disk:



If you select a menu item (disk or folder), it is displayed in a new system window.
The **Copy the path** command copies the complete pathname as text into the clipboard, using the separators of the current platform.

- **"Licenses" Folder**
The **"Licenses" Folder** button displays the contents of the active Licenses folder in a new system window.
All the license files installed in your 4D environment are grouped together in this folder, placed on your hard disk. When they are opened with a Web browser, these files display information concerning the licenses they contain and their characteristics.

The **Licenses** folder is found at the following location:

- Under Windows: *Documents and Settings\All Users\Application Data\4D\Licenses*
- Under Mac OS: *Library:Application Support:4D:Licenses*

The location of this folder can vary depending on the version or language of your operating system.

Note You can also access this folder from the "Update License" dialog box (available in the **Help** menu).

When you click on the **Licenses" Folder** button, **Licenses** folder in a standard system window. If you have activated your 4D application, the folder must contain at least one license file (file in html format). If you have activated several products, plug-ins or extensions, the folder will contain several license files.

Tables

This page lists all the tables of the database (including invisible tables) as well as their characteristics: number of each table, total number of records, fields and indexes for each table.

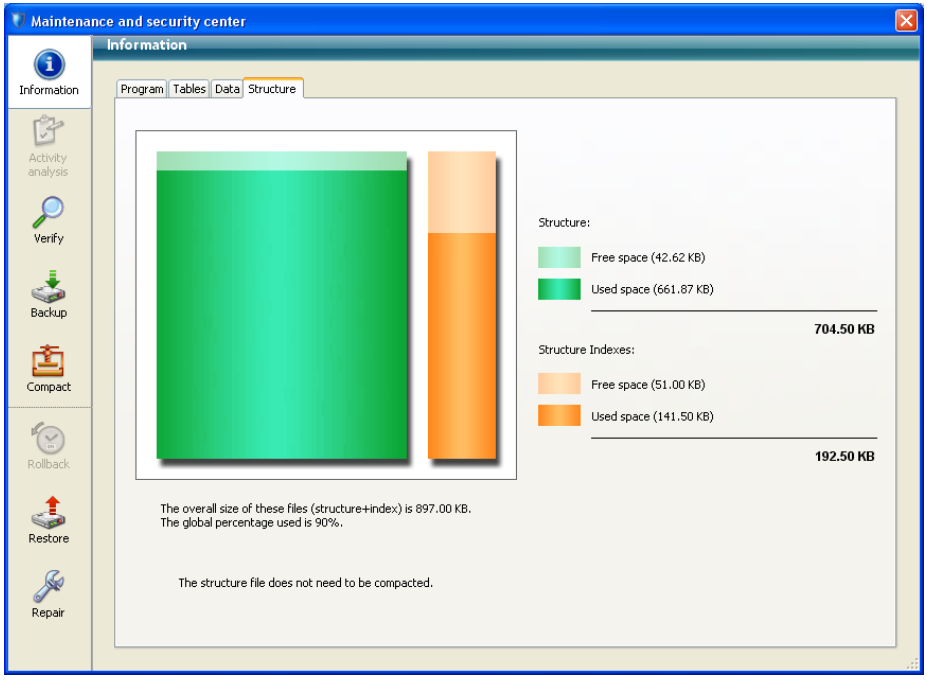
Data and Structure

The **Data** and **Structure** pages provide information about the fill and occupation rates of the database structure and data files.

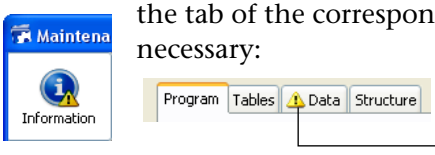
Notes

- These pages cannot be accessed in Maintenance mode.
- The **Structure** page is only available in the 4D Developer and 4D Server applications.

This information is provided in graph form:



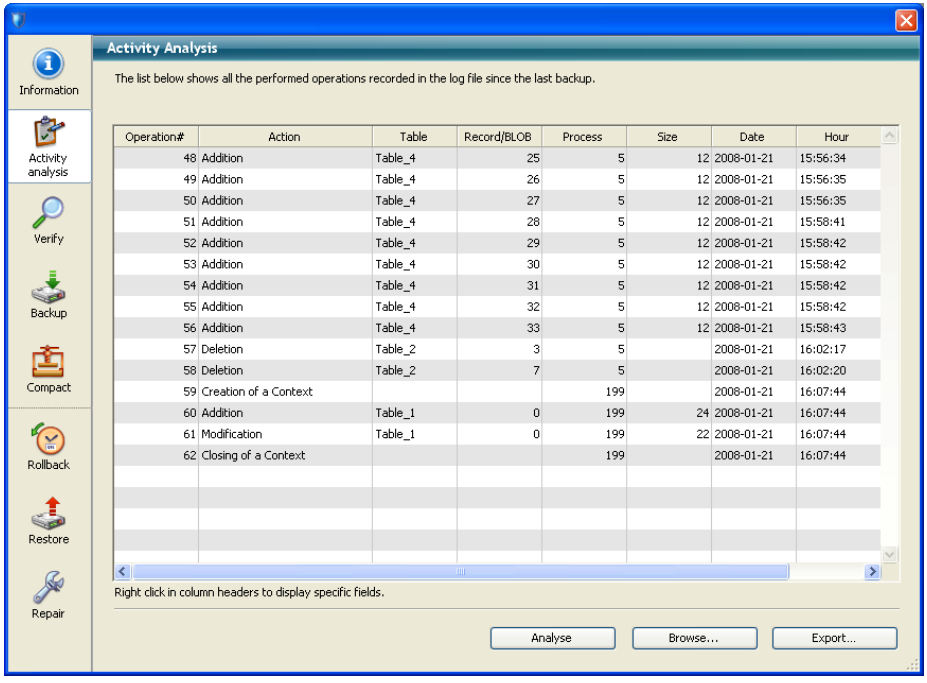
Files that are too fragmented reduce disk and thus database performance. If the occupation rate is too low, 4D will indicate this by a warning icon (which is displayed on the **Information** button and on the tab of the corresponding file type) and specify that compacting is necessary:



A warning icon is also displayed on the button of the **Compacting** page (see the paragraph “Compact” on page 1221) :

Activity Analysis

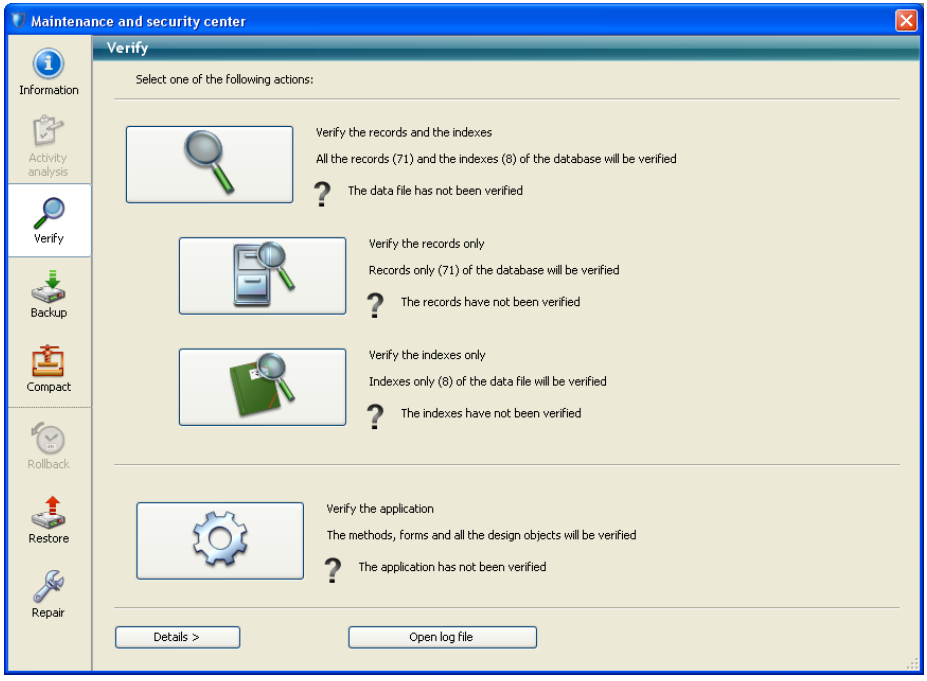
This page is used to view all the operations found in the log file.



For more information about this page, please refer to the paragraph “Parsing a Log File” on page 1193.

Verify

This page allows you to verify data and structural integrity. The verification can be carried out on records and/or indexes as well as on design objects (methods, forms, and so on).



Note This feature only checks the data integrity. If errors are found and repairs are needed, you will be advised to use the **Repair** page (see the paragraph “Repair” on page 1227).

Actions

The page contains four action buttons that can be used for direct access to the verification functions:

- **Verify the records and the indexes**: Starts the total data verification procedure.

- **Verify the records only**: Starts the verification procedure for records only (indexes are not verified).
- **Verify the indexes only**: Starts the verification procedure for indexes only (records are not verified).

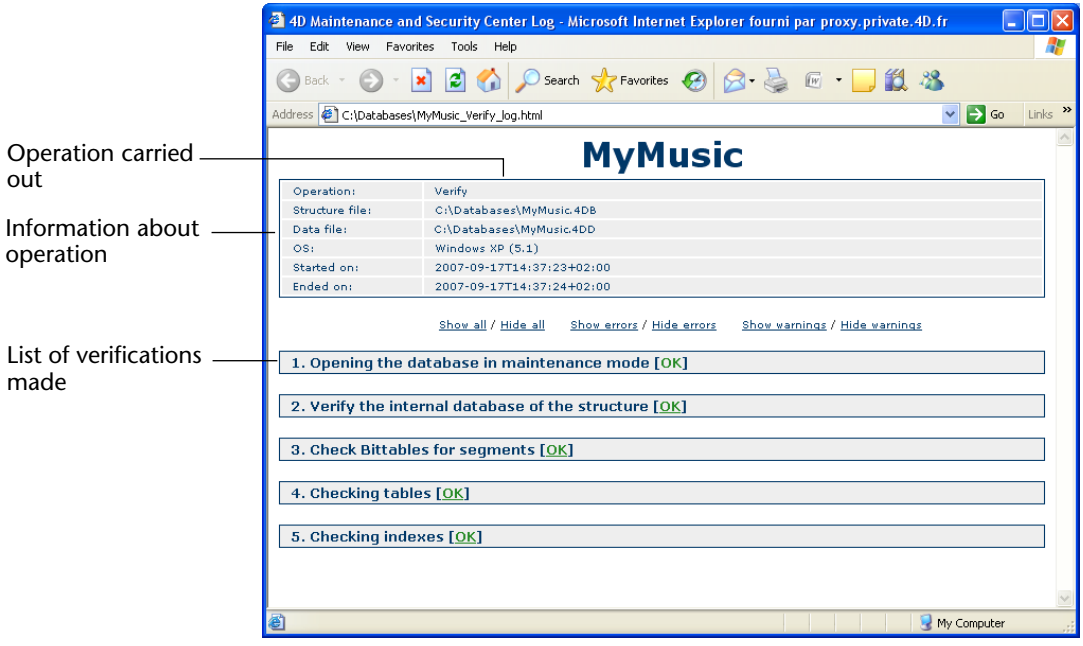
Note Verification of records and indexes can also be carried out in detail mode, table by table (see the paragraph “Details” on page 1218).

- **Verify the application**: Starts the verification procedure for all the objects defined in the Design mode (tables, methods, forms, and so on).

Open Log File

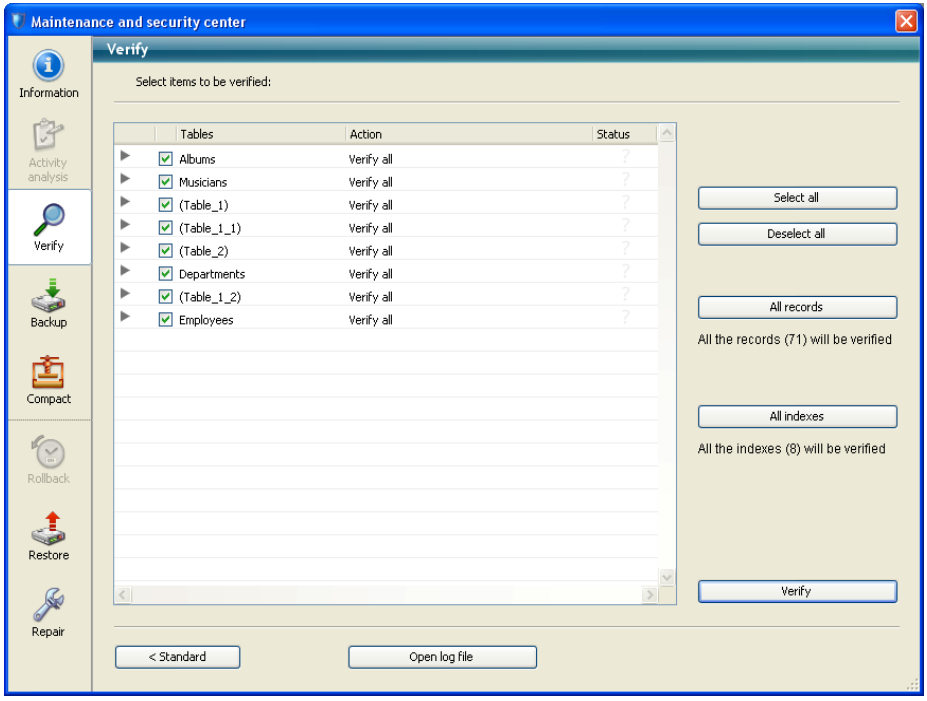
Regardless of the verification requested, 4D generates a log file in the database folder. This file lists all the verifications carried out and indicates any errors encountered, when applicable ([OK] is displayed when the verification is correct). It is created in XML and HTML format and is named “*DatabaseName_Verify_log*”.

When you click on the **Open log file** button, 4D displays the log file in HTML format in the default browser of the machine:



Details

The **Details** button displays a detailed page that can be used to view and select the actual records and indexes to be checked:

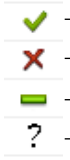


Specifying the items to be verified lets you save time during the verification procedure.

The main list displays all the tables of the database. For each table, you can limit the verification to the records and/or indexes. Click on the triangle-shaped icon to expand the contents of a table or the indexed fields and select/deselect the checkboxes as desired. By default, everything is selected.
You can also use the **Select all**, **Deselect all**, **All records** and **All indexes** shortcut buttons.

For each row of the table, the “Action” column indicates the operations to be carried out. When the table is expanded, the “Records” and “Indexed fields” rows indicate the number of items concerned.

The Status column displays the verification status of each item using symbols:

 — Verification carried out with no problem
— Verification carried out, problems encountered
— Verification partially carried out
— Verification not carried out

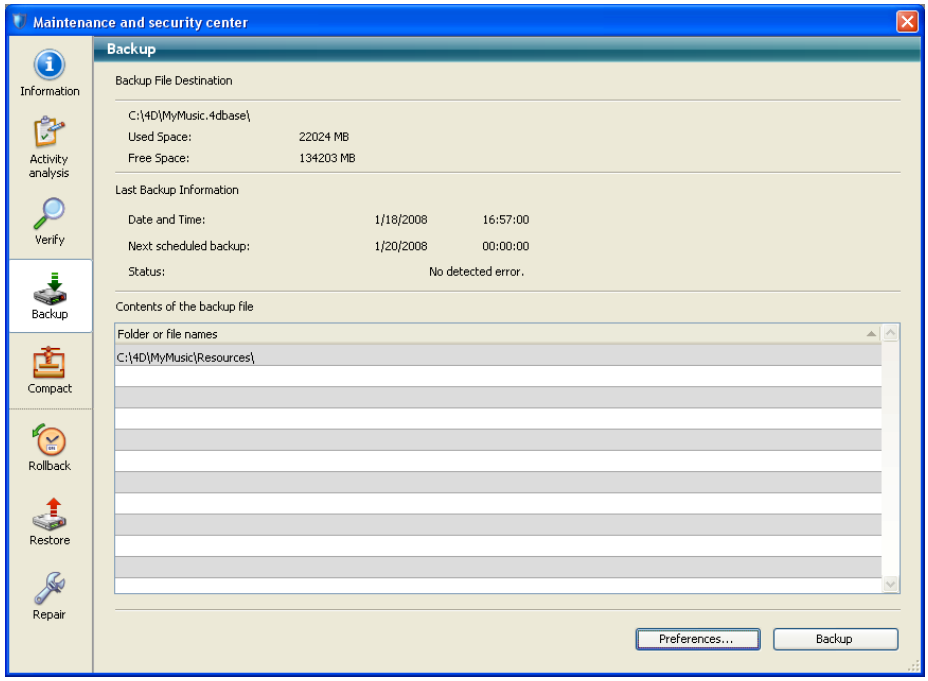
Click on **Verify** to begin the verification or on **<Standard** to go back to the standard page.

The **Open log file** button can be used to display the log file in HTML format in the default browser of the machine (see the paragraph "Open Log File" on page 1217).

Note The standard page will not take any modifications made on the detailed page into account: when you click on a verification button on the standard page, all the items are verified. On the other hand, the settings made on the detailed page are kept from one session to another.

Backup

The **Backup** page is used to view the backup parameters of the database and to launch a manual backup:



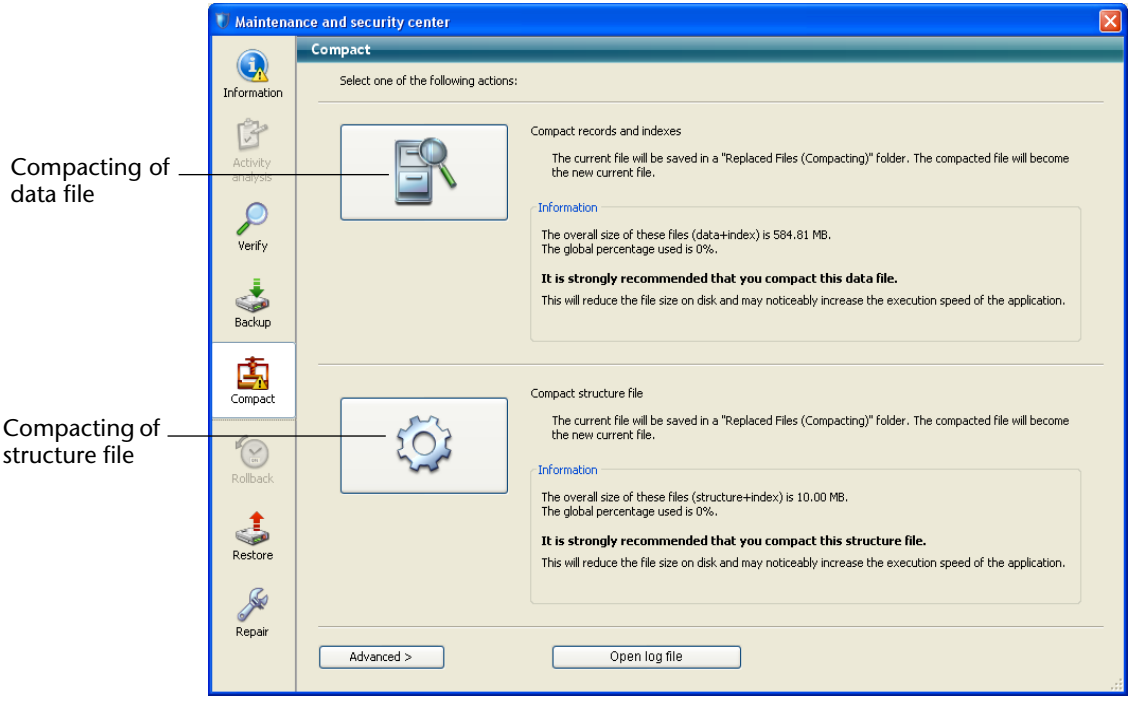
- **Backup File Destination**
  This area displays information about the location of the database backup file. It also indicates the free/used space on the backup disk.
- **Last Backup Information**
  This area provides the date and time of the last backup (automatic or manual) carried out on the database.
- **Contents of the backup file**
  This area lists the files and folders included in the backup file.

The **Backup** button is used to launch a manual backup.

This page cannot be used to modify the backup parameters. To do this, you must click on the **Preferences...** button. For more information, please refer to the paragraph “Performing a Backup” on page 1170.

Compact

This page is used to access the data and structure file compacting functions.



Why Compact Your Files?

The files may contain unused spaces (“holes”). In fact, when you delete records, forms, and so on, the space that they occupied previously in the file becomes empty. 4D reuses these empty spaces whenever possible, but since data size is variable, successive deletions or modifications will inevitably generate unusable space for the program. The same goes when a large quantity of data has just been deleted: the empty spaces remain unassigned in the file.

The ratio between the size of the data file and the space actually used for the data is the *occupation* rate of the data. A rate that is too low can lead, in addition to a waste of space, to the deterioration of database performance. Compacting can be used to reorganize and optimize storage of the data in order to remove the “holes”.

The “Information” areas summarize the data concerning the fragmentation of the files and suggests operations to be carried out. The “Data” and “Structure” tabs on the “Information” page of the MSC indicate the current fragmentation of the database files (see the paragraph “Information” on page 1212).

Note Compacting is only available in maintenance mode. If you attempt to carry out this operation in standard mode, a warning dialog box will tell you that the database will be closed and restarted in maintenance mode.
It is possible to compact a data file that is not opened by the database (see the paragraph “Compact Records and Indexes” on page 1224).

Compacting the Data or Structure File

The standard compacting procedure for the data and structure file are identical.

To directly begin the compacting of the data or structure file, click on the corresponding button in the MSC window:

Data Structure

Note Since compacting includes the duplication of the original file, the button is disabled when there is not adequate space available on the disk containing the file.

This operation compacts the main file as well as any index files.
4D copies the original files and puts them in a folder named “**Replaced Files (Compacting)**”, which is created next to the original file. When the operation is completed, the compacted files automatically replace the original files. The database is immediately operational without any further manipulation.

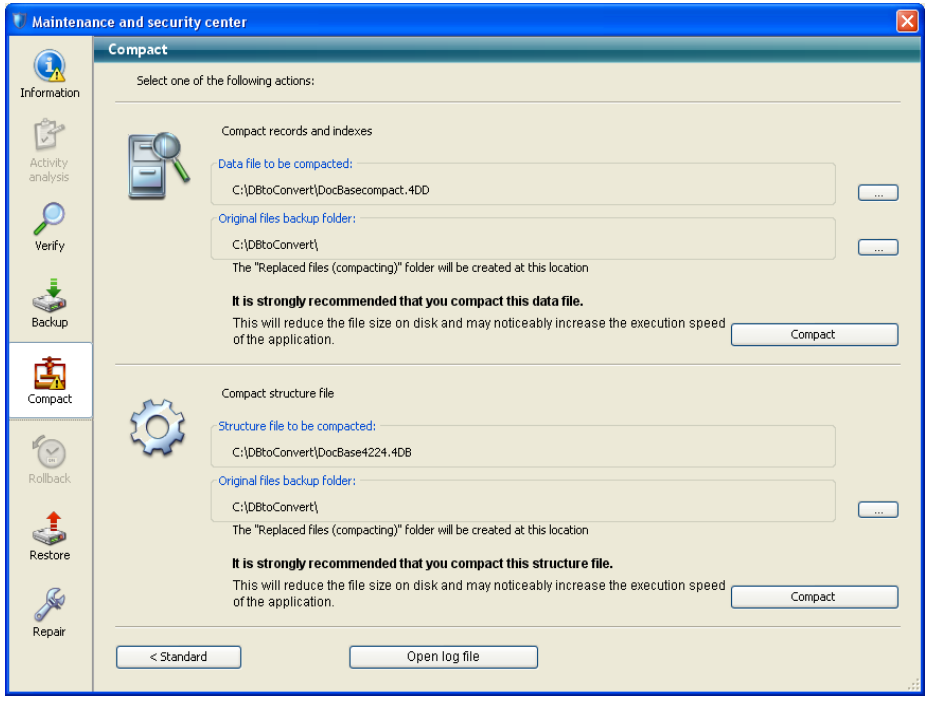
Notes • You can modify the folder where the original files are saved using the advanced mode (see the paragraph “Advanced Mode” on page 1223).
• If you have carried out several compacting operations, a new folder is created each time. It will be named “Replaced Files (Compacting)_1”, “Replaced Files (Compacting)_2”, and so on.

Each compacting operation involves the duplication of the original file which increases the size of the application folder. It is important to take this into account (especially under Mac OS where 4D applications appear as packages) so that the size of the application does not increase excessively. Manually removing the copies of the original file inside the package can be useful in order to keep the package size down.

After compacting is completed, 4D generates a log file in the database folder. This file, named *databasename_compact_log*, is created in XML and HTML format. It can be used to view all of the operations carried out.
When you click on the **Open log file** button, 4D displays the log file in HTML format in the default browser of the machine (see the paragraph "Open Log File" on page 1217).

Advanced Mode

The Compacting page contains an **Advanced>** button, which can be used to access an options page for compacting data and structure files:



Compact Records and Indexes

The “Compact records and indexes” displays the pathname of the current data file as well as a [...] button that can be used to specify another data file. When you click on this button, a standard Open document dialog box is displayed so that you can designate the data file to be compacted. You must select a data file that is compatible with the open structure file.
Once this dialog box has been validated, the pathname of the file to be compacted is indicated in the window.

The second [...] button can be used to specify another location for the original files to be saved before the compacting operation. This option can be used more particularly when compacting voluminous files while using different disks.

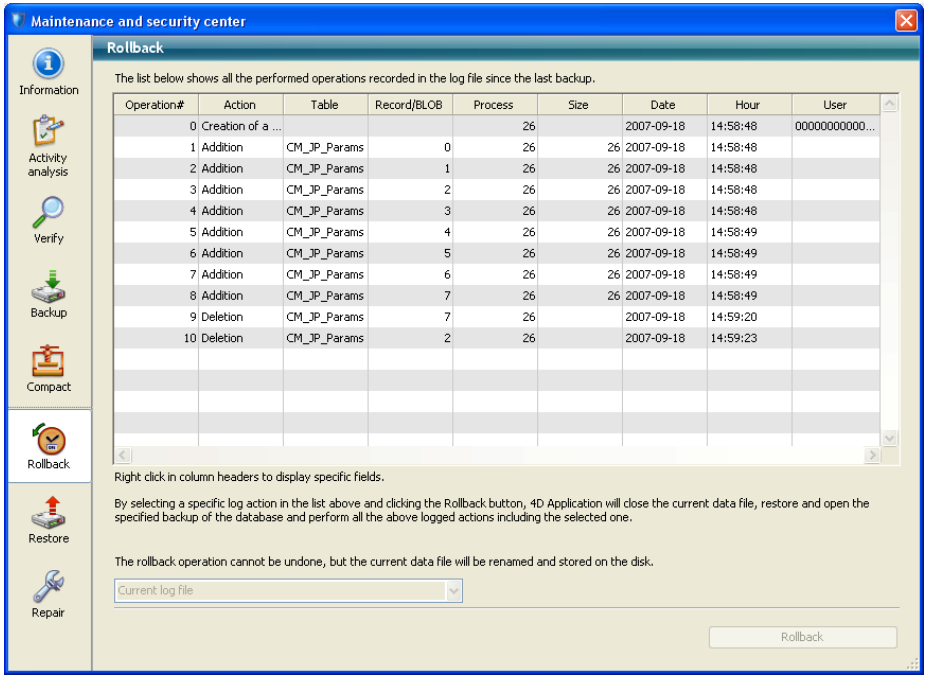
Compact Structure File

The “Compact structure file” area displays the pathname of the database structure file as well as a [...] button that can be used to specify another location for the original files to be saved before compacting begins.

When you click on one of the **Compact** buttons, the operation begins immediately. When compacting is complete, 4D generates a report in the database folder. When you click on the **Open log file** button, 4D displays the log file in HTML format in the default browser of the machine (see the paragraph “Open Log File” on page 1217).

Rollback

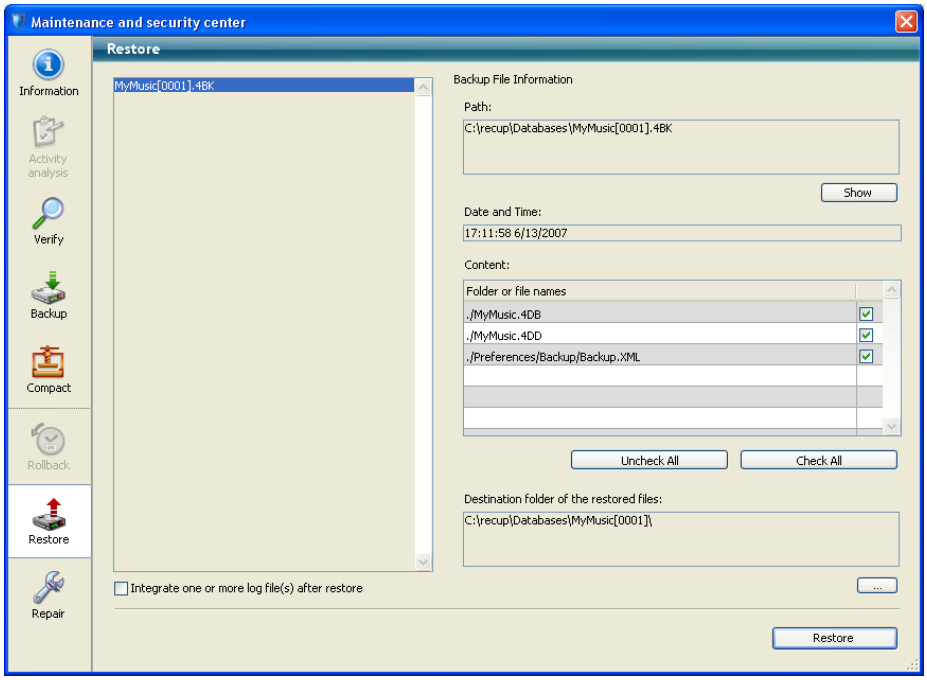
This page is used to access the rollback function among the operations carried out on the data file. This function is only available when the database functions with a log file. It resembles an undo function applied over several levels. It is particularly useful when a record has been deleted by mistake in a database.



For more information about this dialog box, please refer to the paragraph “Undoing Operations” on page 1204.

Restore

This page is used to view and manually restore database backups:



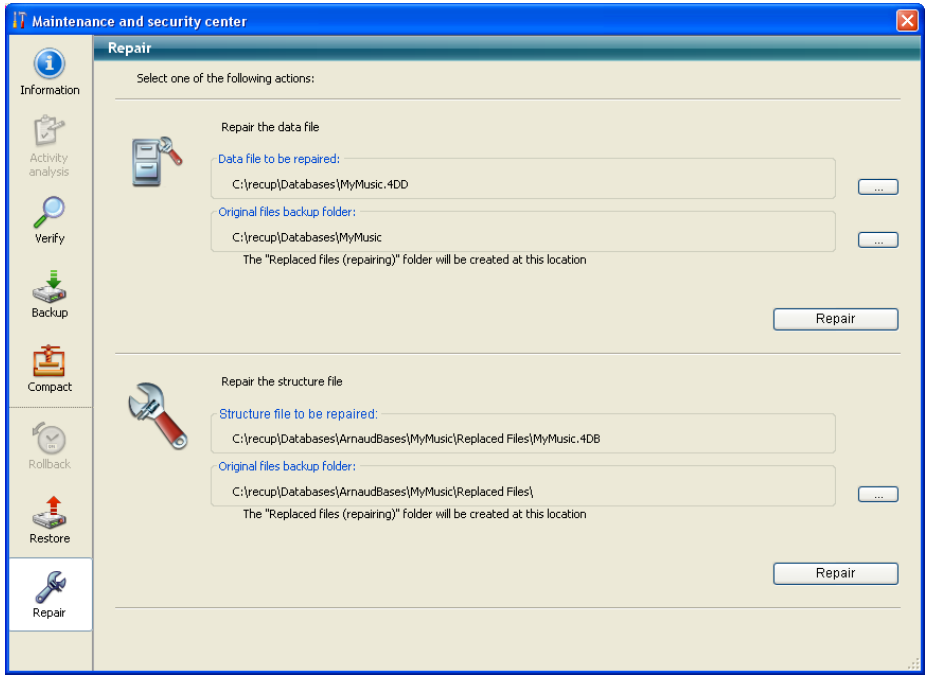
The list in the left-hand part of the window displays the existing database backups. When you select a backup in this list, the right-hand part of the window displays information about this particular backup.

For more information about this dialog box, please refer to the paragraph “Manually Restoring a Backup (MSC)” on page 1202.

Repair

This page is used to repair the data or structure file when it has been damaged. Generally, you will only use these functions at the request of 4D, when anomalies were detected when opening the database or following a verification.

Note For more information about verification, please refer to the paragraph “Verify” on page 1216.



Each repair operation involves the duplication of the original file which increases the size of the application folder. It is important to take this into account (especially under Mac OS where 4D applications appear as packages) so that the size of the application does not increase excessively. Manually removing the copies of the original file inside the package can be useful in order to keep the package size down.

Note Repairing is only available in maintenance mode. If you attempt to carry out this operation in standard mode, a warning dialog box will inform you that the database will be closed then restarted in maintenance mode.

Repair the Data File

The “Repair the data file” area displays the pathname of the current data file as well as a [...] button that can be used to specify another data file. When you click on this button, a standard Open document dialog box is displayed so that you can designate the data file to be repaired. You must select a data file that is compatible with the open structure file.
Once this dialog box has been validated, the pathname of the file to be repaired is indicated in the window.

By default, the original data file will be duplicated before the repairing operation. It will be placed in a subfolder named “Replaced files (repairing)” in the database folder. The second [...] button can be used to specify another location for the original files to be saved before repairing begins. This option can be used more particularly when repairing voluminous files while using different disks.

Repair the Structure File

The “Repair the structure file” area displays the pathname of the database structure file as well as a [...] button that can be used to specify another location for the original files to be saved before repairing them.

When you click on one of the **Repair** buttons, if necessary the database is closed then reopened in maintenance mode before the operation begins.

After the repair operation is completed, 4D generates a log file in the database folder. This file, named *databasename_Repair_log*, is created in XML and HTML format. It can be used to view all of the operations carried out. When you click on the **Open log file** button, 4D displays the log file in HTML format in the default browser of the machine (see the paragraph “Open Log File” on page 1217).

26 Compilation

It is possible to compile your application, i.e., translate all of your methods into machine language. Compiling a database lets you check the consistency of the code and accelerate its execution, as well as making it possible to protect the code in its entirety. Compilation is an indispensable step between the development of databases using 4D and their deployment as stand-alone applications.

The compilation process is entirely automatic; however, compilation requires greater rigor when writing 4D code. The "Compiler Commands" section of the *4D Language Reference* manual provides advice and specific information concerning programming with a view to compilation. Furthermore, keep in mind that the compiler will indicate any programming errors and situate them in their context.

Introduction

This paragraph describes the theory of operation and the advantages of database compilation.

What is a Compiler?

The computer is a device where commands are written using only "0"s and "1"s. This binary language is called *machine language*. The heart of the machine, the microprocessor, understands only this language. A program written in any high-level computer language (C, C++, SQL, Java, BASIC, 4D, and so on) is first translated into machine language, so as to be understandable to the computer's microprocessor.

There are two ways to do this:

- The statements can be translated during execution; the program is then said to be *interpreted*.
- The statements are translated as a whole before program execution; the program is then said to be *compiled*.

Interpreted Mode

When a series of statements is executed using an interpreter, the process can be broken down as follows:

- The program reads a statement in the program's own language,
- It translates the statement into machine language,
- It executes the statement.

This cycle is executed for each of the statements in the program. The program that handles the execution of this kind of cycle is called the *interpreter*. For a database in the process of development, 4D methods are interpreted.

Compiled Mode

A compiled program is translated in its entirety prior to execution. This process results in a new file that contains a set of statements in machine language. This set is saved for repeated use—the translation is performed only once and the compiled version of the program is available for repeated execution.

This phase is completely independent from any use of the program. The program that handles the translation is called the *compiler*.

Compiled Objects

The compiler in 4D compiles the database methods, project methods, triggers, form methods and object methods in your database. If you do not have any of these elements in an application, the compiler will have nothing to compile.

When you have successfully completed compilation, the use of the compiled database is identical to that of the original one.

PPC and Intel

The compiler in 4D can generate code for PPC (Mac OS) or Intel (Mac Intel or Windows) processors. You can choose to generate code for either one or both of these platforms.

When both types of code are requested (**Compile for PPC and Intel processors** option in the Preferences), the compiler generates "fat binary" code, i.e., both PowerPC code and Intel code. When the database is used in compiled mode, the code suited for the processor of the user machine will be run.

If the compiled database must be used with 4D Server, the appropriate code will be loaded and run on each of the client machines: PPC or Intel on Macintosh and Intel on PC Windows.

Why Compile Your Database?

The first benefit of compilation is, of course, speed of execution. There are two further benefits directly linked to compilation:

- Systematic code checking,
- Database and component protection.

Speed of Execution

The increased speed is due to two characteristics of compiled code: direct code translation, once and for all, and direct access to variable and method addresses.

- **Direct and final code translation**
  The code of the methods written in 4D will be translated once and for all using the compiler. The time required in interpreted mode to translate all the statements is saved whenever you use a compiled database.

  Here is a simple case that illustrates this point. Take the case of a loop containing a sequence of statements that is repeated 50 times:

```
For (i;1;50)
   `Sequence of statements
End for
```

  In an interpreted database, each statement in the sequence is translated 50 times. Using the compiler eliminates the translation phase for each statement. For every statement in the sequence, we save 50 translations.

- **Direct access to variable and method addresses**
  In interpreted databases, variables are accessed through a name. Therefore, 4D must access the name in order to obtain the variable's value.

  In the compiled code, the compiler attaches an address to each variable, writes the variable's address directly in the code, and goes directly to that address whenever necessary.

Notes
- Operations requiring disk access may not be affected because their speed of execution is limited by the rate of transmission between the computer and its peripherals (drive or hard disk).
- Comments are not translated so they do not appear in the compiled code. Therefore, comments do not affect the execution time in compiled mode.

Checking Your Code

The compiler also operates as a syntax checker for your databases. It systematically checks your code and notes possible ambiguities, whereas 4D only does this when the method is executed.

Suppose that one of your methods contains a series of tests as well as a sequence of statements to be executed. It is unlikely that you would fully test for all cases if the number of tests was very large. Therefore, a syntax error in an untested case might not show up until an end user encounters the case.

This sort of problem is avoided when you use a compiled database. When you compile a database, the compiler scans the entire database and analyzes each statement. The compiler detects any abnormality and generates an error message or warning.

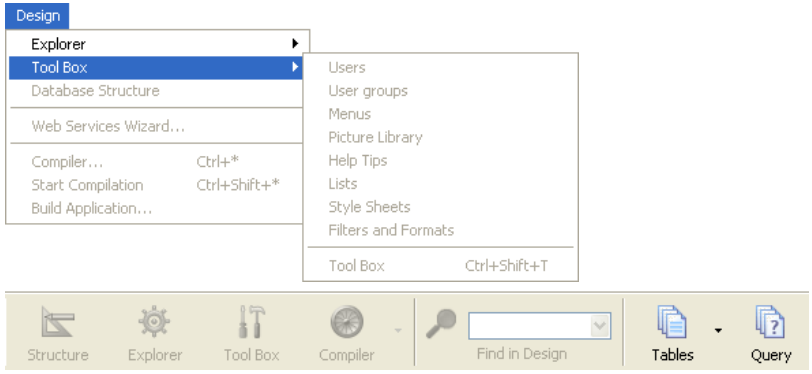
Protecting Your Applications and Components

Once you have compiled your database, you can use the application builder to erase the interpreted code. In this case, access to the Design environment (except for records) is blocked.

Note For more information about the application builder, refer to the chapter "Finalizing and Deployment of 4D Applications" on page 1251.

In a compiled database, the commands related to development are disabled:

Commands disabled in a compiled database



When a compiled component is installed into a host database, the shared project methods are accessible in the Explorer and can be called in methods of the host database but their contents do not appear in the preview area nor in the debugger. The other project methods of the component never appear.

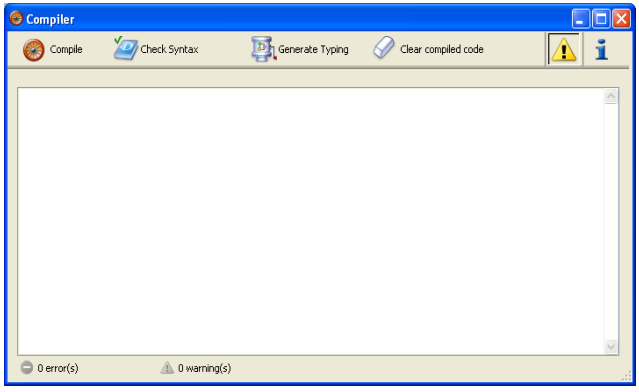
Note For more information about components, please refer to the chapter "4D Components" on page 1269.

The benefits are:

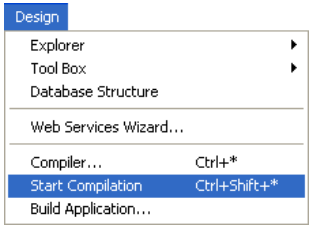
- The database development cannot be modified, intentionally or by accident,
- Your methods are now protected.

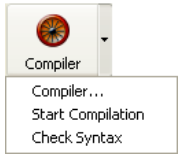
Compilation in 4D

A compiler is integrated into 4D. Database compilation is carried out using the following dialog box:



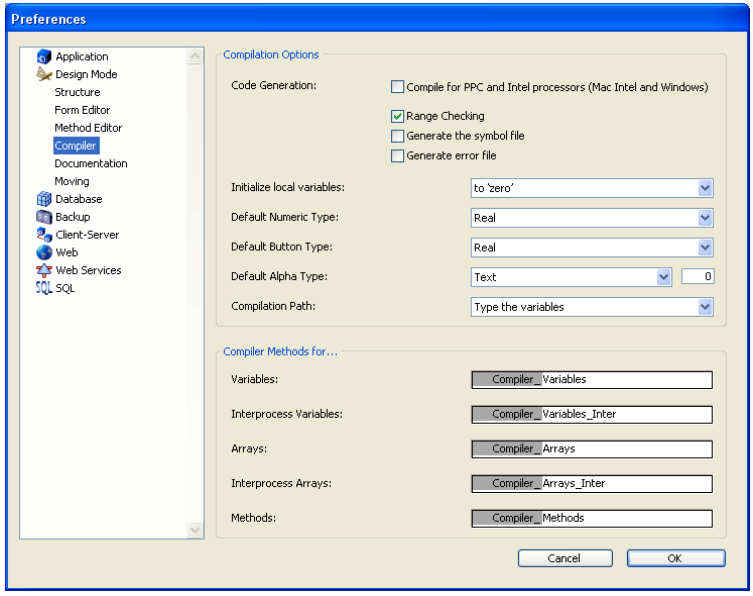
It is also possible to launch compilation directly using the current settings via the **Start Compilation** command found in the **Design** menu and in the menu associated with the “Compiler” button of the tool bar:



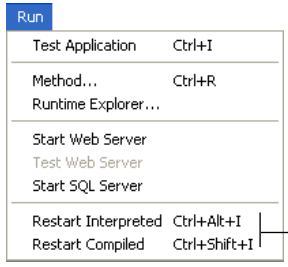


For more information about compilation, please refer to the “Compile” paragraph on page 1237.

Compilation is carried out in keeping with generic compilation options, set in the Preferences dialog box:



Once the database is compiled, it is still possible to switch from interpreted mode to compiled mode, and vice versa, at any time via the **Run** menu, without having to quit the 4D application — except when the interpreted code has been erased (see the previous paragraph).



Options for restarting the application

The Open database dialog box also allows the choice of interpreted or compiled mode on startup of the database (see the "Open Dialog Box Options" paragraph on page 40).

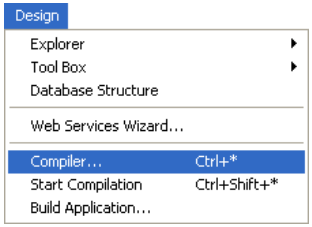
Note If you modify the structure of your database in interpreted mode, you must recompile it in order to have them taken into account in compiled mode.

When you switch from one mode to the other, 4D closes the current mode and opens the new one. This amounts to exiting then reopening the application. Consequently, each time you change from one mode to another, 4D executes the two following database methods (if specified) in this order:

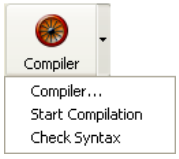
- On Exit Database Method
- On Startup Database Method

Compiler Window

To display the compiler window, select the **Compiler...** command in the **Design** menu:



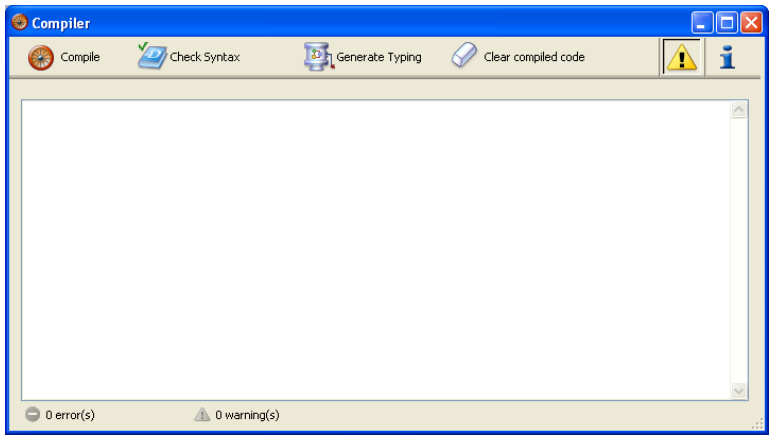
You can also click on the "Compiler" button of the 4D tool bar or choose the **Compiler** command in the associated menu:



Note These commands are disabled if the database does not contain at least one method.

This window is used for launching the compilation of the database or checking the syntax of the methods.

In addition, buttons can be used for generating/regenerating database typing methods, erasing the compiled code, displaying or hiding the warnings, and accessing the Preferences directly.



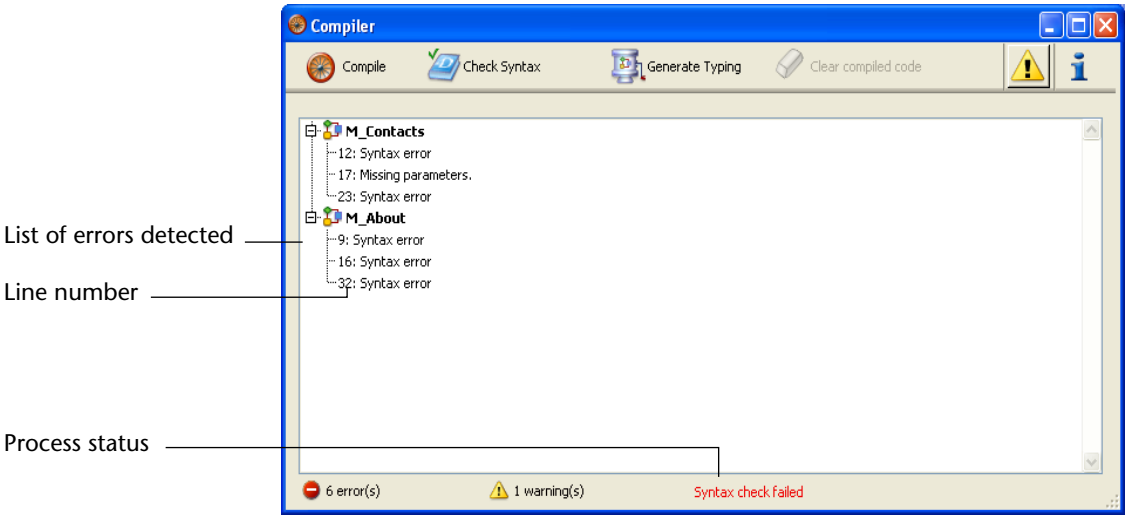
Notes Database compilation requires an appropriate license. Without this license, it is not possible to carry out a compilation (the **Start Compilation** command and the **Compile** button are disabled). Nevertheless, it is still possible to check the syntax and generate typing methods.

The commands of this window are described below.

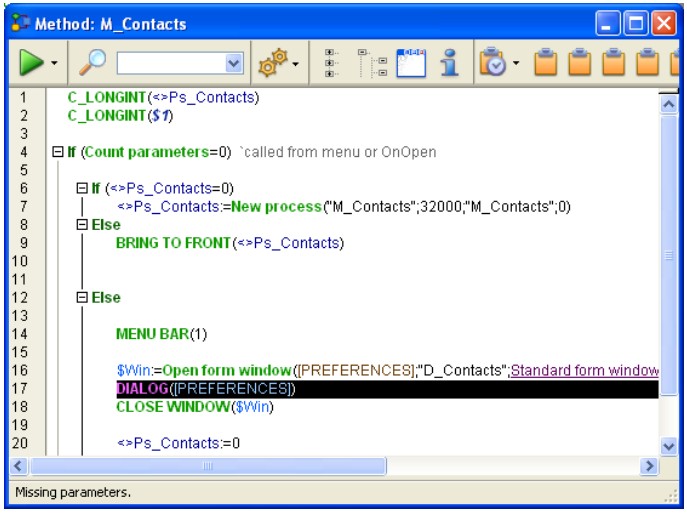
Compile

This button causes the immediate launching of the database compilation process. It is the exact equivalent of the **Start Compilation** command in the **Design** menu.If the database has already been compiled, the new code compiled will replace the former.
Initially, different passes are carried out for checking, typing and initialization, in accordance with the configuration set in the Preferences window (see the “Compiler Page” paragraph on page 186).

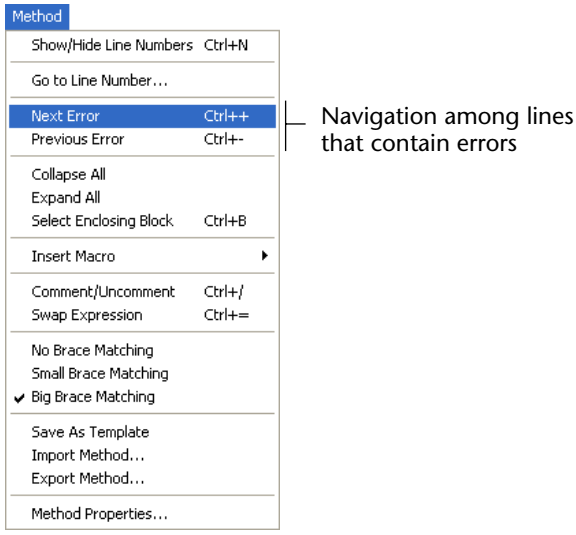
If no errors are detected, the actual compilation begins.
If errors are detected, the process is stopped and the information area of the window displays the method names and line numbers concerned in a hierarchical list:



Double-click on each error detected in order to open the method concerned directly in the 4D Method editor; the line containing the error is highlighted and the type of error is displayed in the syntax area of the window:



The **Previous Error** / **Next Error** commands of the **Method** menu of the editor allow you to navigate among the lines containing errors:



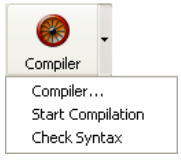
Note The number of errors found during your first compilations may be daunting, but do not let this put you off. You will soon discover that they often spring from the same source, i.e., non-compliance with certain database conventions. The compiler always provides a precise diagnosis of the errors in order to help you correct them.

Check Syntax

This button starts the execution of the syntax-checking phase. At the end of checking, any errors detected are listed in the information area. As described in the following paragraph, it is possible to double–click on an error line in order to display the corresponding method.

This option is the only one available if you do not have a suitable license to allow the compilation of applications.

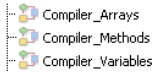
Syntax checking can also be launched directly using the **Check Syntax** command associated with the “Compiler” button:



Generate Typing

This button creates (or updates) the typing "compiler methods." Compiler methods are project methods that group together all the variable typing declarations, process and interprocess arrays, as well as the local variable declaration methods. These methods, when they exist, are used directly by the compiler during code compilation, which accelerates compilation. If these methods already exist, their contents are updated.

These methods, whose names must mandatorily begin with "Compiler_", are generated by 4D. You can set the default name for each of the 5 compiler methods in the Preferences (see the "Compiler Methods" paragraph on page 189). The compiler methods generated and maintained by 4D automatically have the "Invisible" attribute:



Only the necessary compiler methods (i.e., those for which items already exist in the database) are generated.

The information area indicates any errors found during method creation or updating. Double-clicking on an error line causes the method and line concerned to be displayed in the Method editor.

For more information about typing methods, refer to the "Compiler Commands" section of the *4D Language Reference* manual.

Clear Compiled Code

This button is used to delete the compiled code of the structure file. When you click on it, all of the code generated during compilation is deleted. The size of the structure file will be reduced accordingly if you carry out a compacting operation (see the "Compact" paragraph on page 1221).

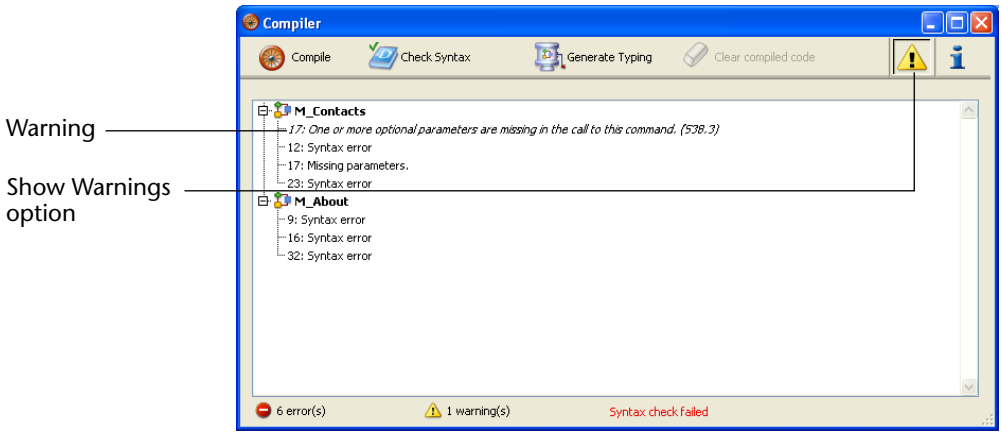
The **Restart Compiled** command of the **Run** menu is then disabled and the "Compiled Database" option of the open database dialog box is grayed out.
Note that generated compiler methods are not deleted by this command.

Show Warnings

Warnings are specific messages generated by the compiler when it checks the syntax. These messages are intended to draw your attention to statements that might lead to execution errors. They do not prevent compilation. For more information about warnings, refer to the "Warnings" paragraph on page 1247.

Depending on circumstances and the programming style used, these warnings may be more or less relevant. The **Show Warnings** option can be used to display/hide the warnings in the information area of the compiler window.

When this option is checked, the warnings (if any) are displayed in the window, after the other error types. They appear in *italics*:



Double-clicking a warning opens the corresponding method.

Disabling Warnings During Compilation

It is possible to selectively disable certain warnings during compilation. To do this, you just need to insert the following in the code of a 4D method:

```
`%W-<warning number>
```

The warning numbers are now specified at the end of each message in the list of compilation errors.

For example, to disable the following warning:
1: Pointer in an array declaration (518.5)

... you just need to write the following comment in a 4D method, preferably a *COMPILER_xxx* method (method compiled first):

```
`%W-518.5
```

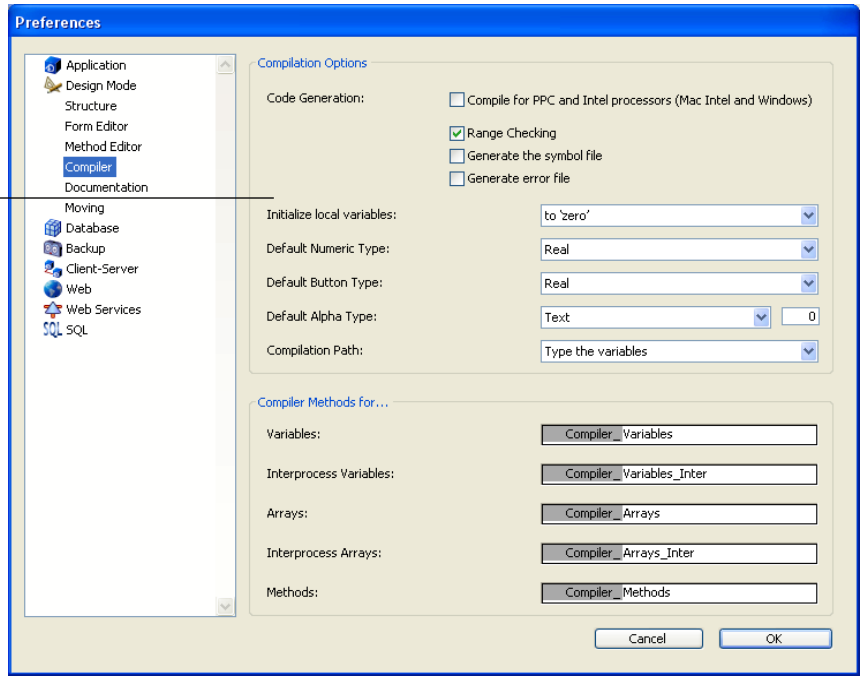
Direct Access to Compilation Preferences

The generic compilation options are set in the Preferences dialog box of the application. Options set in this dialog box will be applied to all the databases opened using the current 4D application.

To access the compilation options, click on the button at the top right of the compiler window:

The **Compiler** page of the “Design Mode” theme then appears:

Generic compilation options



The compilation preferences are described in the “Compiler Page” paragraph on page 186.

Compilation Diagnostic Aids

There are three types of aids for the analysis and correction of databases:

- The actual analysis aid is provided by the symbol file. This table lets you find your way through your variables quickly. It is a valuable tool for interpreting the error messages reported by the compiler.

- The correction aid is provided by the error file which you can use as a text file.
- The execution aid or *range checking* provides you with an additional tool for monitoring the consistency and reliability of your applications.

Note Significant assistance is also provided for the typing of variables by the automatic compiler methods — see the "Generate Typing" paragraph on page 1240.

Symbol File

The symbol file is a text type document whose length will depend on the size of your databases. By default, this file is not generated at the time of the compilation. To do so, you must check the corresponding option in the application (see the "Compiler Page" paragraph on page 186). When it is generated, the file is placed in the folder containing the database structure and is named:

- Under Windows, DatabaseName.sym,
- Under Mac OS, DatabaseName.symb.

The symbol file is displayed as follows when it is opened using a text editor:

```
4D_Airports2003         10/9/2003       5:34 PM

<>4DCOMMENTS    Text            (M)  Compiler_Variables_Inter
<>4DQUERY       Text            (M)  Compiler_Variables_Inter
<>4DQUERYCOMMENTS       Text            (M)  Compiler_Variables_Inter
<>4DTABLETITLE  Text            (M)  Compiler_Variables_Inter
<>AAIRCRAFTDISTINCTRANGEKM      Long integer    1 dimension     (M)
Compiler_Arrays_Inter
<>AAIRCRAFTFAMILY       Text    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTID   Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTMANUFACTURER Text    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTNAME Text    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTRANGEKM      Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTSPEED        Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRPORTGROUPID       Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRPORTIATACODE      Fixed string:3  1 dimension     (M)  PS_InitArrays
<>AAIRPORTID    Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRPORTLEVEL Long integer    1 dimension     (M)  GetCitiesOfContry
<>ACAPITALAIRPORTS1     Text    1 dimension     (M)  Compiler_Arrays_Inter
<>ACAPITALAIRPORTS2     Text    1 dimension     (M)  Compiler_Arrays_Inter
<>ACAPITALAIRPORTSID1   Long integer    1 dimension     (M)
Compiler_Arrays_Inter
<>ACAPITALAIRPORTSID2   Long integer    1 dimension     (M)
Compiler_Arrays_Inter
<>ACITY Text    1 dimension     (M)  PS_InitArrays
<>ACOMPANYID    Long integer    1 dimension     (M)  PS_InitArrays
<>ACOMPANYNAME  Text    1 dimension     (M)  PS_InitArrays
<>ACOUNTRY      Text    1 dimension     (M)  PS_InitArrays
<>ACOUNTRYCITY  Text    1 dimension     (M)  GetCitiesOfContry
<>ACOUNTRYCITYID        Long integer    1 dimension     (M)  GetCitiesOfContry
```

The header displays the name of the database and the date and time of the document creation.

The document is divided into four parts:

- List of interprocess variables.
- List of process variables.
- List of local variables, in their method.
- Complete list of project methods and database methods with their parameters, if applicable.

List of Process and Interprocess Variables

These two lists are divided into four columns:

- The first column contains the names of process and interprocess variables and arrays used in your database. These variables are listed in alphabetical order.
- The second column contains the type of the variable. Types are set by compiler directive commands or are determined by the compiler based on the use of the variable. If the type of a variable cannot be determined, the column is empty.
- The third column lists the number of dimensions if the variable is an array.
- The fourth column contains a reference to the context in which the compiler established the type of the variable. If the variable is used in several contexts, the context mentioned is the one used by the compiler to determine its type.
  - If the variable was found in a database method, the database method name is given as it has been defined in 4D, preceded by **(M)***.
  - If the variable was found in a project method, the method is identified as it has been defined in 4D, preceded by **(M)**.
  - If the variable was found in a trigger (table method), the table name is given, preceded by **(TM)**.
  - If the variable was found in a form method, the form name is given, preceded by the table name and **(FM)**.
  - If the variable was found in an object method, the object method's name is given, preceded by the form name, table name, and by **(OM)**.

- If the variable is an object in a form and does not appear in any project, form or object methods, nor any triggers, the name of the form in which it appears is given, preceded by **(F)**.

At the end of each list, you can find the sizes of the process and interprocess variables in bytes.

Note When compiling, the compiler cannot determine in which process a given process variable is used. A process variable can have a different value in each process. Consequently, all process variables are systematically duplicated as each new process is launched: it is thus advisable to watch out for the amount of memory that they will take up. Also, keep in mind that the space for process variables is not related to the stack size for the process.

List of Local Variables

The list of local variables is sorted by database method, project method, trigger (table method), form method, and object method, in the same order as in 4D.

This list is divided into three columns:

- The first column contains the list of local variables used in the method;
- The second column contains the type of the variable;
- The third column lists the number of dimensions if the variable is an array.

Complete List of Methods

A complete list of your database and project methods is given at the end of the file, with the data types of their parameters and the returned result.

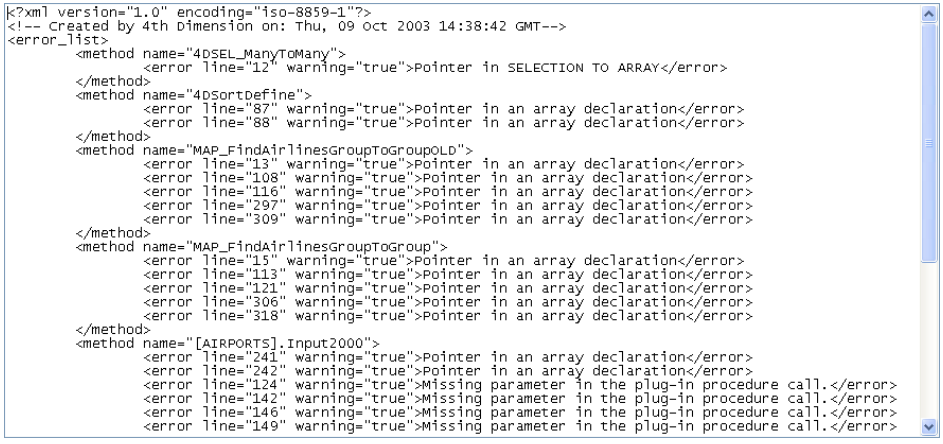
This information is presented in the following format:
Method name(parameter data types):result data type.

Error File

You can choose whether or not to generate an error file during compilation using an option located in the Preferences of the application (see the "Compiler Page" paragraph on page 186). When it is generated, the error file is automatically named DatabaseName_errors.xml and is created next to the structure file of the database.

Although the errors can be accessed directly via the compiler window, it can be useful to have an error file that can be transmitted from one machine to another, particularly in the case of several different developers working together in a client-server environment. The error file is generated in XML format in order to facilitate automatic parsing of its contents. It also allows the creation of customized error display interfaces.

The length of the error file depends on the number of errors and warnings issued by the compiler. When you open an error file using a text editor, it looks like this:



```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- Created by 4th Dimension on: Thu, 09 Oct 2003 14:38:42 GMT-->
<error_list>
        <method name="4DSEL_ManyToMany">
                <error line="12" warning="true">Pointer in SELECTION TO ARRAY</error>
        </method>
        <method name="4DSortDefine">
                <error line="87" warning="true">Pointer in an array declaration</error>
                <error line="88" warning="true">Pointer in an array declaration</error>
        </method>
        <method name="MAP_FindAirlinesGroupToGroupOLD">
                <error line="13" warning="true">Pointer in an array declaration</error>
                <error line="108" warning="true">Pointer in an array declaration</error>
                <error line="116" warning="true">Pointer in an array declaration</error>
                <error line="297" warning="true">Pointer in an array declaration</error>
                <error line="309" warning="true">Pointer in an array declaration</error>
        </method>
        <method name="MAP_FindAirlinesGroupToGroup">
                <error line="15" warning="true">Pointer in an array declaration</error>
                <error line="113" warning="true">Pointer in an array declaration</error>
                <error line="121" warning="true">Pointer in an array declaration</error>
                <error line="306" warning="true">Pointer in an array declaration</error>
                <error line="318" warning="true">Pointer in an array declaration</error>
        </method>
        <method name="[AIRPORTS].Input2000">
                <error line="241" warning="true">Pointer in an array declaration</error>
                <error line="242" warning="true">Pointer in an array declaration</error>
                <error line="124" warning="true">Missing parameter in the plug-in procedure call.</error>
                <error line="142" warning="true">Missing parameter in the plug-in procedure call.</error>
                <error line="146" warning="true">Missing parameter in the plug-in procedure call.</error>
                <error line="149" warning="true">Missing parameter in the plug-in procedure call.</error>
```

The structure of the error file is as follows:

- At the top of the file is the list of errors and warnings, sorted by method and in the order of their creation in 4D;
- In the ***General errors*** section, all the typing impossibilities and identity ambiguities are grouped together.

  These errors and warnings are listed using the following format:
  - First, the line number in the method (0 indicates general errors);
  - Second, the warning attribute indicates whether the detected anomaly is a warning (warning="true") or an error (warning="false");
  - And third, a diagnostic that describes the error.

  If your database does not have any general errors, the file will not have a “General errors” section.

An error file may contain three types of messages:

- Errors linked to a specific line,
- General errors,
- Warnings.

Errors Linked to a Specific Line

These errors are displayed in context — the line in which they were found — with an explanation. The compiler reports this type of error when it encounters an expression in which it sees an inconsistency related to data type or syntax.

In the compiler window, double–click on each error detected in order to open the method concerned directly in the 4D Method editor with the line containing the error highlighted.

The list of syntax/typing diagnostic errors is found in the *4D Language Reference* manual.

General Errors

These are errors that make it impossible to compile the database. There are two cases in which the compiler reports a general error:

- The data type of a process variable could not be determined.
- Two different kinds of objects have the same name.

General errors are so named because they cannot be linked to any specific method. In the first case, the compiler could not perform a specified typing anywhere in the database. In the second, it was unable to decide whether to associate a given name with one object rather than with another.

The list of general errors is found in the *4D Language Reference* manual.

Warnings

Warnings are not errors. Warnings do not prevent the database from being compiled; they simply point out potential code errors.

In the compiler window, warnings appear in *italics*. Double–click on each warning to open the method concerned directly in the 4D Method editor with the line concerned by the warning highlighted.

The list of warnings is found in the *4D Language Reference* manual.

It is possible to disable certain warnings (see the "Disabling Warnings During Compilation" paragraph on page 1241).

Range Checking

Range checking is checked by default in the Preferences of the application (see the "Compiler Page" paragraph on page 186).

Whereas all the other options operate during the compilation process, range checking begins when you run the compiled database. That is, range checking messages only appear when your compiled database is running.

Range checking provides additional analysis with respect to the quest for logical and syntactical consistency which normally characterizes a compiler. During range checking, the compiler poses the following question: "Considering what you have requested, will the result that I am likely to obtain surprise you?". Range checking is an "in situ" controller; it evaluates the status of objects in the database at a given time.

Here is how range checking works. Suppose that you declared the array MyArray as Text. The number of elements in MyArray may vary depending on the current method. If you want to assign the value "Hello" to element 5 of MyArray, you would write: MyArray{5}:="Hello". If MyArray has five or more elements at that time, everything is fine. Assignment proceeds normally. However, if MyArray has less than five elements at that time, your assignment no longer makes sense.

A situation like this cannot be detected at the time of compilation since it presupposes the execution of the methods. The compiler would not know the circumstance in which this method is called. Only range checking enables you to monitor what is actually happening while your database is in use. In the above example, the compiler would display an execution error from within 4D. It is easy to see why range checking is especially valuable when arrays, pointers, and strings of characters are being processed.

The messages sent by the compiler when you request range checking are listed in the *4D Language Reference* manual.

Disabling Range Checking Locally

Even when range checking has been enabled, there may be some cases where you prefer that it not be applied to certain parts of code that are considered to be reliable. More particularly, in the case of loops that are repeated a great number of times, and when running the compiled database on older machines, range checking can significantly slow down processing. Insofar as you have the *certitude* that the code concerned is reliable and cannot cause system errors, you can disable range checking locally.

To do this, you must surround the code to be excluded from range checking with the special comments `%R- and `%R+.

The `%R- comment disables range checking and `%R+ enables it again:

```
... `Range checking is enabled
`%R-
... ` Place the code to be excluded from range checking here
`%R+
... `Range checking is enabled again for the rest of the method
```

Note This mechanism will only operate when range checking is enabled.

Diagnosing Anomalies

Suppose you notice anomalies when running your databases. Before you start speculating about the possible sources of these problems, remember the assistance provided by the compiler.

Potential anomalies are:

- 4D displays its own error messages.
If possible, correct errors in your database according to instructions provided by 4D. If these are too general, compile your database again, making sure that the Range Checking option is enabled. Retest your database. At the location where the 4D message was displayed, you will see a more informative message from the compiler.
- Your compiled database does not perform exactly like your interpreted database. Take a closer look at the warning messages.
- Number or String variables do not return expected values. Check the default typing options in the Preferences and examine the symbol file to check that all your variables are typed properly.
- Your database works in interpreted mode, but the application crashes in compiled mode. Make sure that you compiled the database using the Range Checking option and check to see whether your compiled database is using the same plug-ins as the ones you used when compiling.

27 Finalizing and Deployment of 4D Applications

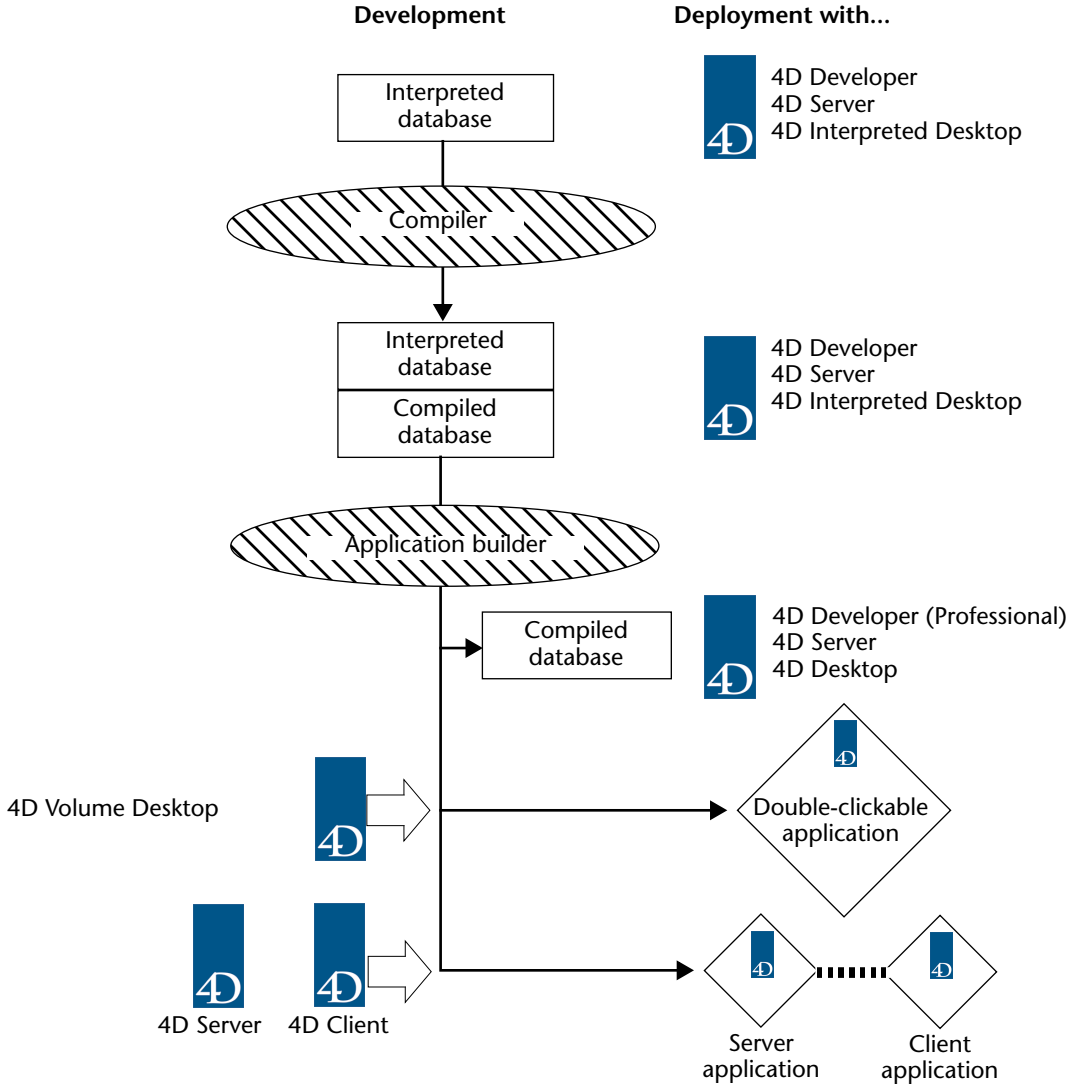
4D includes a final application builder. This builder simplifies the finalization and deployment process for 4D compiled applications. It automatically handles the specific features of different operating systems, in particular the building of software packages under Mac OS, and facilitates the deployment of client-server applications.

The application builder allows you to:

- Build a compiled database, without interpreted code,
- Build a stand-alone, double-clickable application, i.e., merged with 4D Volume Desktop, the 4D database engine,
- Build different applications from the same compiled database via an XML project,
- Build homogeneous client-server applications,
- Build client-server applications with automatic updating of client parts.

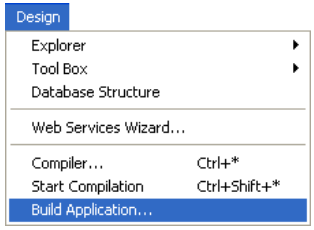
4D Server Application building is available only in the 4D single-user version.

Deployment possibilities for 4D databases are summarized in the following diagram:



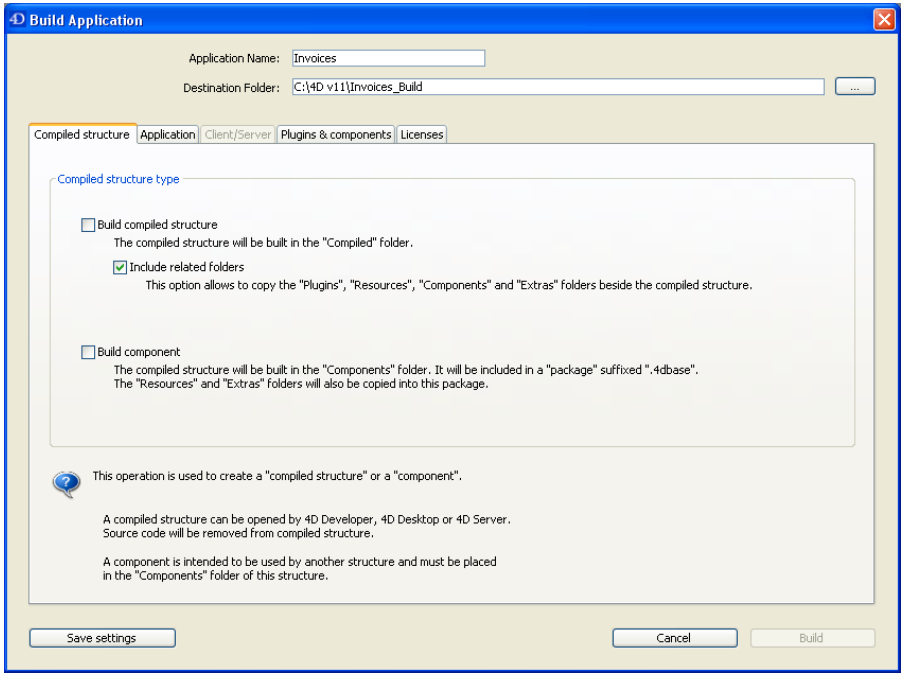
Application Builder

Application building is carried out using the Build Application window. To display this window, select the **Build Application...** command in the **Design** menu of 4D:



Note Building can only be carried out once the database is compiled. If you select this command without having previously compiled the database, or if the compiled code does not correspond to the interpreted code, a warning dialog box appears indicating that the database must be (re)compiled.

The application building window appears. It includes several pages that can be accessed using tabs:



- The **Compiled structure** tab can be used to build a compiled version only of the structure or a component.
- The **Application** tab can be used to set and build a single-user version of the application.
- The **Client-Server** tab allows you to set the client-server version of the application.
- The **Plug-ins & components** tab allows you to set which plug-ins and components to integrate into the application.
- The **Licenses** tab allows you to set the serial numbers to integrate.

You can simultaneously set different "target" parameters (single-user and client-server) for the application; each must be explicitly selected using a check box.

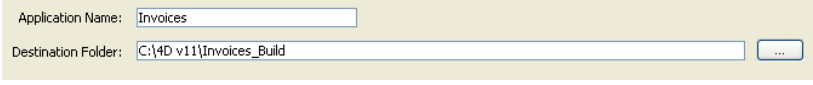
The **Build** button builds the applications corresponding to all selected targets. When you click this button, 4D displays a progress bar indicating the different phases being executed. The new parameters, if valid, will be saved in the application project (see the "XML Keys of Parameters" paragraph on page 1255).

The **Save settings** button lets you record the parameters set without launching the application build.

If you click on **Cancel** or if an error occurs, the files being generated are deleted and a warning dialog box informs you of the cause of the interruption.

Definition of the Application Name and Destination

The upper part of the application building window allows you to set the name and location of the files to be generated.



The "Application Name" area contains, by default, the name of the database structure file. This name will be used for the generated files (compiled database, component, double-clickable application and client-server application). 4D will automatically add the necessary suffixes (.4dc, .exe, server...) according to the type of application built.

When you keep the default name, the name of the application reflects any modification made to the database structure file name.

If you modify the name of the application, the new name is used by default for each new build of the current database. The entered name must therefore NOT have an extension. In addition, it must not contain any characters forbidden by the operating system (such as ".?!" under Windows, ":" under Mac OS, etc.).

The "Destination Folder" area is used to indicate the location where the generated items will be placed. By default, 4D uses a folder named "*StructureName*_Build" placed next to the folder (or package under Mac OS) of the database selected.

To modify the destination folder, click on the selection button [...] located to the right of the display area. A folder selection dialog box appears enabling you to indicate the new destination folder. Once this dialog box is validated, the complete pathname of the folder is displayed. The new location will be used by default for each subsequent build of the current database.

At the time of building, 4D will automatically create one or more intermediary folders (entitled "Compiled Database," "Components," "Final Application," "Client Application" or "Server Application" according to the type of build requested) in the specified location. This avoids the risk of accidentally deleting files with the same name and enables several types of builds to be performed simultaneously.

XML Keys of Parameters

Each parameter of this window is stored as an XML key in the application project file named "BuildApp.XML" that is placed in the BuildApp subfolder of the database Preferences folder.

Default parameters are used the first time the dialog box is used. The contents of the project file are updated, if necessary, when you click **Build** or **Save settings**.

You can define several other XML projects for the same database and employ them using the BUILD APPLICATION command (for more information about this command, refer to the *4D Language Reference* manual).

Please note that XML keys provide additional options besides those displayed in the Build Application dialog box, notably:

- Setting an IP address or the port number of the server.
- Setting an range of compatible version numbers between the client application and the server application (for example, client applications 1.1 to 1.3 can connect to server application 1.3).
- Setting a pathname for the data file, which can be used for building applications that are immediately operational, without it being necessary to designate a data file on the user machine.

The description of these keys are detailed in a separate documentation called *4D XML Keys-BuildApplication*.

Log File

4D generates a log file in XML format the first time an application is built. A log file is generated for each application project.
The log file is named "ProjectName.log.xml" (*ProjectName* is the name of the application project, i.e. BuildApp by default) and is placed next to the application project file.

The log file stores the following information for each build:

- The start and end of building of targets.
- The name and full access path of the files generated.
- The date and time of the build.
- Any errors that occurred.

Building a Compiled Structure or Component

The **Compiled structure** page of the application builder lets you build a standard compiled structure file and/or a compiled component.

In both cases, the compiled structure files generated are identical. Only the architecture of the folders generated differs.

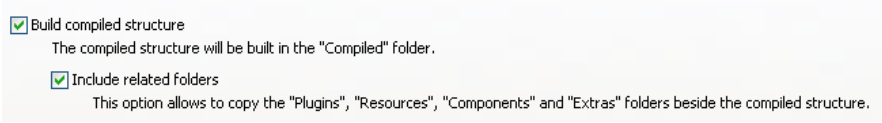
During building, 4D automatically creates the various folders intended to receive the generated files. No matter which option is chosen, the current database is not modified: copies are generated on your disk.

When the compiled structure files include both PPC and Intel[1] code, the corresponding versions of the compiled databases will be built.

Once you have configured the various options of this window, click the **Build** button in order to generate the desired files in the location indicated.

Building a Compiled Structure

This option of the application building window is used to build a database containing only compiled code.



If you have defined “MyDatabase” in the “Application Name” area, you obtain:

- A file named MyDatabase.4dc (compiled database file)
- A file named MyDatabase.4DIndy (structure index file).

The .4dc file can be opened by 4D Developer, 4D Server or 4D SQL Desktop. During the build, 4D deletes the interpreted code of the structure file. It is no longer possible to access the Design mode of this database.

The compiled database is placed:

- In a folder of the type “*MyDatabase*.4dbase,” which has specific properties under Mac OS (see the “.4dbase Extension” paragraph on page 45) — *MyDatabase* is the specified “Application Name,”
- Which is itself placed in a folder named **Compiled Database**,
- Which is itself placed in the specified “Destination Folder.”

Prior to rebuilding a compiled database, 4D erases the previous contents of the “Compiled Database” folder. If you want to keep intermediate versions, you must therefore rename the application or move any compiled versions and/or additional items that you want to keep.

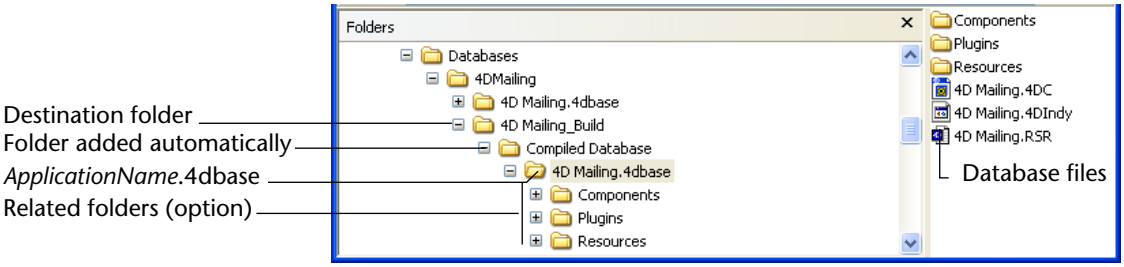
Include Related Folders

When you check this option, any folders related to the database are copied into the *Compiled Database* folder: “Plugins,” “Resources,” “Components” and “Extras.” For more information about these folders, please refer to the “4D Database Files” paragraph on page 42.

1. This option is set in the Preferences of the application (see the “Compiler Page” paragraph on page 186).

This option lets you build “ready to use” compiled structure files.

The typical architecture of a compiled structure is as follows:



Building a Component

This option can be used to build a compiled component from the structure:

A component is a standard 4D structure file in which specific functionalities have been developed. Once the component has been configured and installed in another 4D database (the host database), its functionalities are accessible from the host database. For more information about components, please refer to the chapter “4D Components” on page 1269.

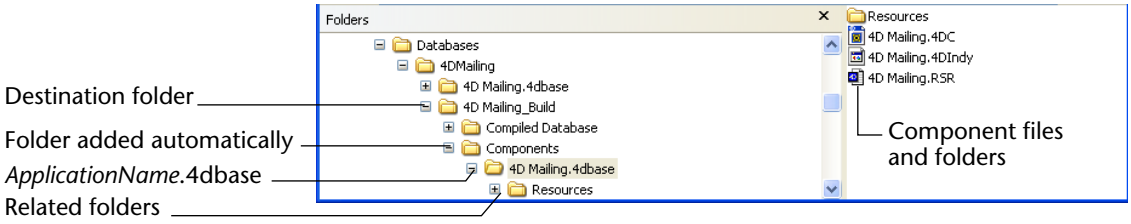
If you have defined “MyComponent” in the “Application Name” area, you obtain:

- A file named MyComponent.4dc (compiled structure file)
- A file named MyComponent.4DIndy (structure index file).

The elements generated are similar to those of a compiled structure, with, however, the following differences:

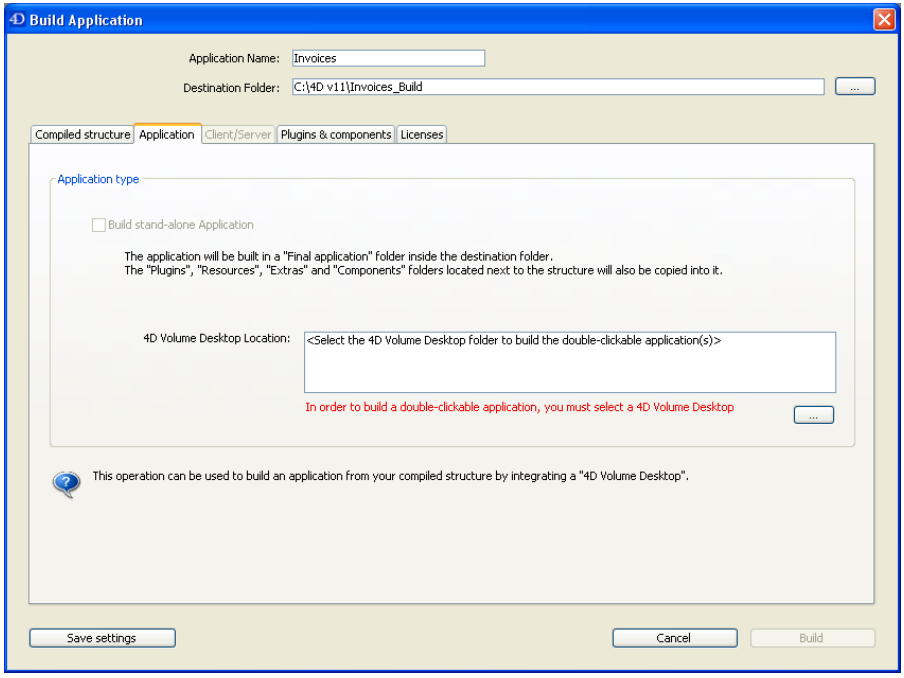
- The component is generated as a folder (or package) in a folder named “Components,”
- The associated “Resources” and “Extras” folder are automatically copied into the folder of the component. On the other hand, any “Components” and/or “Plugins” folders are not copied (a component cannot use plug-ins or other components).

The typical architecture of a compiled structure is as follows:



Building a Double-clickable Application

4D enables you to build a double-clickable application directly from your database. You just need to have 4D Volume Desktop, the 4D database engine, and an appropriate license. To do so, you must display the **Application** page of the application builder:



Double-clickable (.exe) versions of your 4D compiled databases can be created directly in 4D using the “Build Stand-alone Application” function.
Under Mac OS, this function handles the creation of software packages.

The principle consists of merging a compiled structure file with 4D Volume Desktop. The functionalities provided by the 4D Volume Desktop file are linked with the product offer to which you have subscribed. For more information about this point, refer to the sales documentation and to the 4D Internet site (http://www.4d.com).

Selection of 4D Volume Desktop Folder

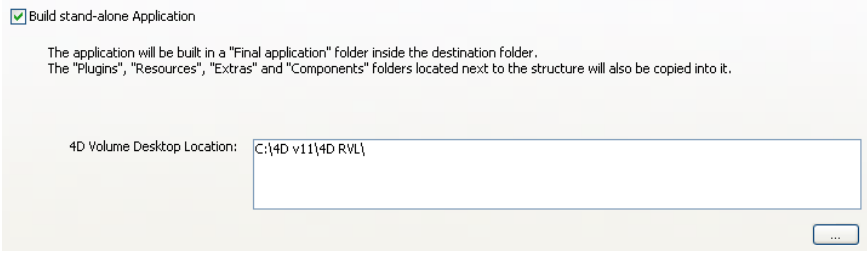
To be able to build a double-clickable application, you must first designate the location of the 4D Volume Desktop folder. The option for building double-clickable applications is grayed out if no folder has been indicated in the corresponding area, or if the folder indicated does not contain a valid 4D Volume Desktop file.

You must select the folder containing the 4D Volume Desktop file.

- Under Windows, the folder contains the 4D Volume Desktop.4DE, 4D Volume Desktop.RSR, as well as various files and folders required for its operation. These items must be placed at the same level as the selected folder.
- Under Mac OS, 4D Volume Desktop is provided in the form of a structured software package containing various generic files and folders.

To select the 4D Volume Desktop folder, click on the [...] button. A dialog box appears allowing you to designate the 4D Volume Desktop folder (Windows) or package (Mac OS).

Once the folder is selected, its complete pathname is displayed and, if it actually contains 4D Volume Desktop, the option for building an executable application is activated:



Generated Files

To build an executable application, click on **Build**.

4D automatically creates a **Final Application** folder in the "Destination Folder" specified and puts a subfolder having the name of the specified application in it.
If you have specified "MyAppli" in the "Application Name" area, you will find the following files in this folder:

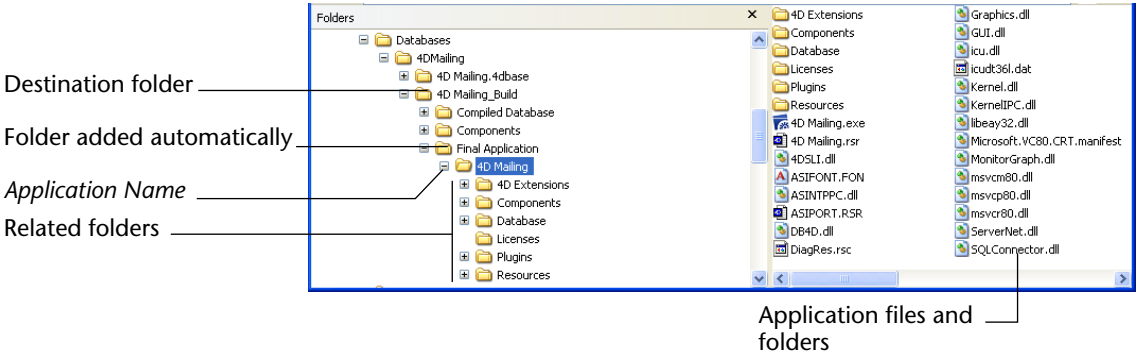
- **Under Windows**
  - MyAppli.exe which is your executable and MyAppli.RSR which contains the application resources.
  - The 4D Extensions and Resources folders, as well as various libraries (DLL) and files necessary for the operation of the application.
  - A Database folder containing more particularly the DatabaseName.4DC and DatabaseName.RSR files making up the compiled structure of the database.
  - (Optional) A Components and a PlugIns folder containing, respectively, any components and/or plug-in files included in the database. For more information about this, refer to the "Plug-in and Component Management" paragraph on page 1264.
  - A Licenses folder containing the list, in the form of an XML file, of license numbers which have been integrated into the application. For more information about this, refer to the "Managing License Numbers" paragraph on page 1265.
  - Any additional items added in the 4D Volume Desktop folder (see the "Customizing the 4D Volume Desktop Folder" paragraph on page 1262).

All these items must be kept in the same folder in order for the executable to operate.

- **Under Mac OS**
  - A software package named MyAppli.app containing your application and all the items necessary for its operation, including the plug-ins, components and licenses. For more information about integrating plug-ins and components, refer to the "Plug-in and Component Management" paragraph on page 1264. For more information about integrating licenses, refer to the "Managing License Numbers" paragraph on page 1265.

Note Under Mac OS, the Application file command of the 4D language returns the pathname of the ApplicationName file (located in the Contents:Mac OS folder of the software package) and not that of the .comp file (Contents:Resources folder of the software package).

The typical architecture of a double-clickable application is as follows (example under Windows)



Customizing the 4D Volume Desktop Folder

When building a double-clickable application, 4D copies the contents of the 4D Volume Desktop folder into the **Final Application** subfolder of the destination folder. You are then able to customize the contents of the original 4D Volume Desktop folder according to your needs.

You can, for instance:

- Install a 4D Volume Desktop version corresponding to a specific language;
- Add a custom PlugIns folder;

- Customize the contents of the 4D Extensions folder.

Note Under Mac OS, 4D Volume Desktop is provided in the form of a software package. In order to modify it, you must first display its contents (**Control+click** on the icon).

Rebuilding a Double-clickable Application

Prior to rebuilding a double-clickable application, 4D erases the contents of the "Final application" folder. Therefore, you must move any versions that you want to keep or change the "Application Name" field before the build.

Location of Web Files

If your double-clickable application is used as a Web server, the files and folders required by the server must be installed in specific locations. These items are the following:

- cert.pem and key.pem files (optional): These files are used for SSL connections and by data encryption commands,
- default Web root folder,
- logweb.txt file (Web requests log).

- **Under Windows**: These items must be installed in the Final Application*MyAppli*\Database subfolder.
- **Under Mac OS**: These items must be installed next to the MyAppli.app software package.

Building a Client-Server Application

4D allows you to build customized client-server applications that are homogenous, cross-platform and with an automatic update option.

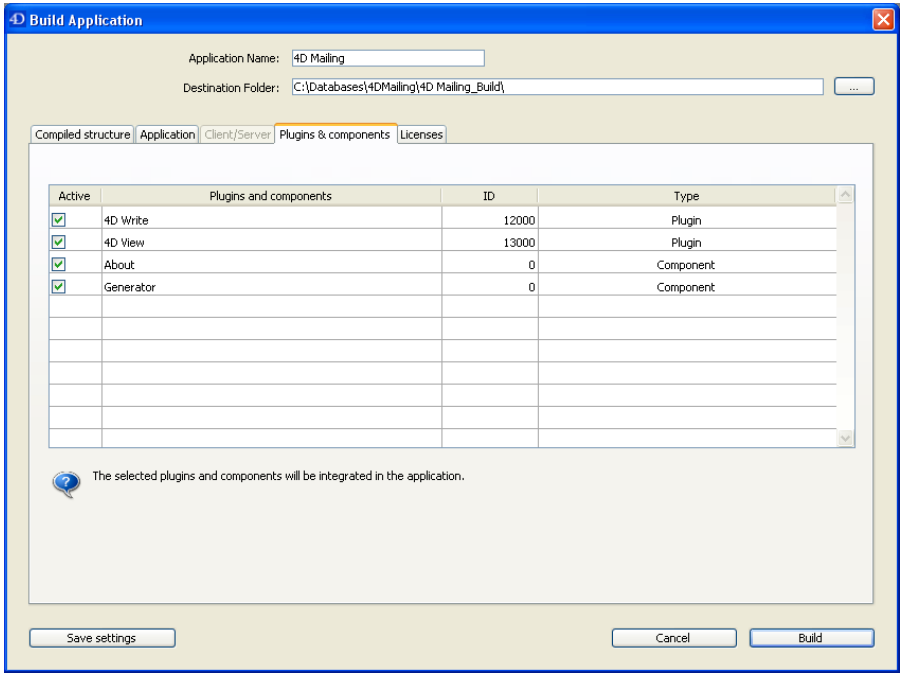
This function is in the process of being implemented in 4D v11 SQL; it is not yet operational at the date this manual was produced. This chapter will be updated upon release of the 4D v11 SQL client/server version.

Plug-in and Component Management

The application builder allows you to set each plug-in and each component that you will use in your single-user or client-server application.

For more information about 4D plug-ins, please refer to the paragraph “Installing Plug-ins” on page 47. For more information about components, please refer to the chapter “4D Components” on page 1269.

To designate the plug-ins and/or components to be integrated, click on the **Plugins & components** tab in order to display the list of items loaded by the current 4D application:



The **Type** column indicates the type of item: plug-in or component. The **Active** column indicates that the items will be integrated into the application built. All the items are checked by default. To deselect a plug-in or a component, click the check box next to it.

If you want to integrate other plug-ins into the executable application, you just need to place them in a PlugIns folder next to the 4D Volume Desktop application or next to the 4D Server application. The mechanism for copying the contents of the source application folder (see the "Customizing the 4D Volume Desktop Folder" paragraph on page 1262) can be used to integrate any type of file into the executable application.

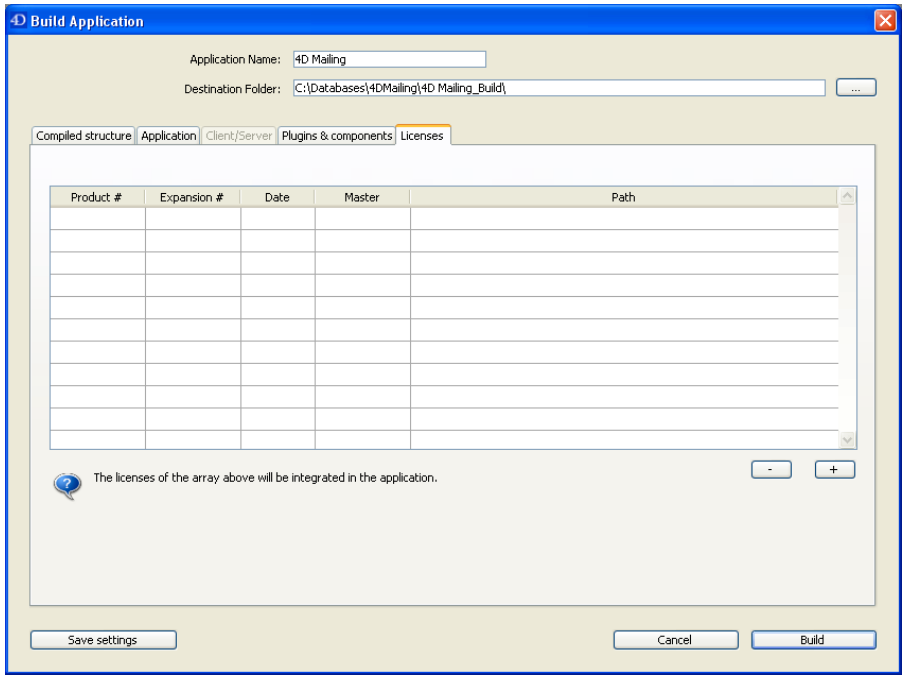
If there is a conflict between two different versions of the same plug-in (one loaded by 4D and the other located in the source application folder), priority goes to the plug-in installed in the 4D Volume Desktop/4D Server folder.

Note The use of plug-ins and/or components in a deployment version requires the necessary license numbers.

Managing License Numbers

The application builder allows you to specify the license number(s) that you want to integrate into your single-user stand-alone application.

Click the **Licenses** tab to display the list of available deployment licenses:



By default, the list is empty. You must explicitly add your *4D Developer Professional* license as well as each *4D Desktop Volume* license to be used in the application built. You can add another *4D Developer Professional* number and its associated licenses other than the one currently being used.

To remove or add a license, use the [+] and [-] buttons on the right-hand side of the window.

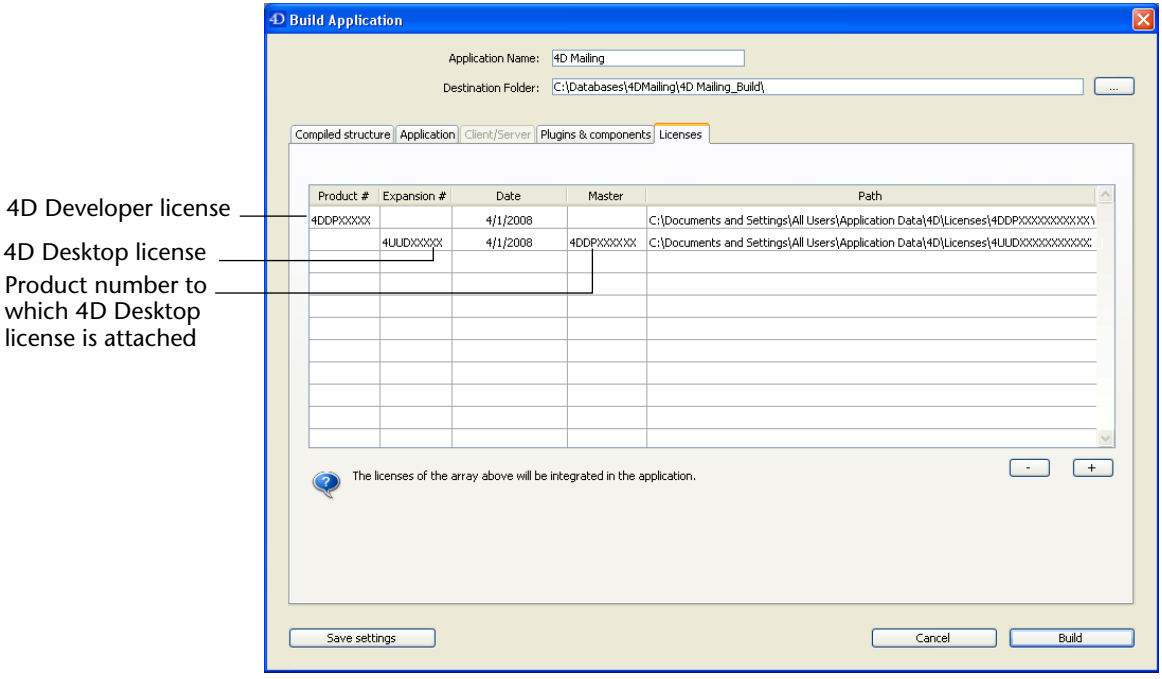
When you click on the [+] button, an open file dialog box appears displaying by default the contents of the [Licenses] folder of your machine. This folder is found at the following location:

- Under Windows: {Disk}*Documents and Settings\All Users\Application Data\4D\Licenses*
- Under Mac OS: {Disk}*Library:Application Support:4D:Licenses*

The location of this folder can vary according to the version or language of your operating system.

You must designate the files that contain your Developer license as well as those containing your deployment licenses. These files were generated or updated when the *4D Developer Professional* license and the *4D Desktop Volume* licenses were purchased.

Once you have selected a file, the list will indicate the characteristics of the license that it contains: product or expansion number, validity date, associated *4D Developer Edition* number (Master column) and location.



If the license is not valid, a message will warn you of this.

You can designate as many valid files as you want. When building an executable application, 4D will use the most appropriate license available.

After the application is built, a new deployment license file is automatically included in the Licenses folder next to the executable application (Windows) or in the package (Mac OS).

Customizing a Stand-Alone Application Icon

4D associates a default icon with double-clickable applications. However, you can customize the icon for each application.

Under Mac OS

When building a double-clickable application, 4D handles the customizing of the icon.
In order to do this, you must carry out the following operation prior to generating the application file:

1 **Create an icon file (icns type) and place it next to the interpreted structure file.**

Note Apple, Inc. provides a specific tool for building icns icon files.

Your icon file must have the same name as the interpreted structure file and include the .icns extension.

4D automatically takes this file into account when building the double-clickable application (the .icns file is renamed ApplicationName.icns and copied into the Resources folder; the *CFBundleFileIcon* entry of the "info.plist" file is updated).

Under Windows

When building a double-clickable application, 4D handles the customizing of its icon.

In order to do this, you must carry out the following operation prior to generating the application file:

1 **Create an icon file (.ico extension) and place it next to the interpreted structure file.**

Your icon file must have the same name as the interpreted structure file and include the .ico extension.

4D automatically takes this file into account when building the double-clickable application.

28 4D Components

A 4D component is a set of 4D objects representing one or more functionalities that can be installed in different databases. For example, you can develop a 4D e-mail component that manages every aspect of sending, receiving and storing e-mails in 4D databases.

Creating and installing 4D components is carried out directly from 4D Developer. Basically, components are managed like plug-ins according to the following principles:

- A component consists of a regular structure file (compiled or not) having the standard architecture or in the form of a package (see the paragraph ".4dbase Extension" on page 45).
- To install a component in a database, you simply need to copy it into the "Components" folder of the database, placed next to the structure file. It is possible to use a shortcut (Windows) or an alias (Mac OS). To uninstall it, simply remove it from the folder.
- It is not possible to include tables in 4D v11 components.

These principles are described in more detail in the following paragraphs.

Note Creating components requires an appropriate license. On the other hand, installing and using components in a 4D application is not restricted.

Definitions

The component management mechanisms in 4D require the implementation of the following terms and concepts:

- **Matrix Database**:4D database used for developing the component. The matrix database is a standard database with no specific attributes. A matrix database forms a single component.
  The matrix database is intended to be copied, compiled or not, into the Components folder of the database that will be using the component (the *host database*, see below).
- **Host Database**: Database in which a component is installed (in the Components folder).
- **Component**: Matrix database, compiled or not, copied into the Components folder of the host database and whose contents are used in the host database.
- **Project Form**: Form not linked with a table. Project forms are particularly suitable for component generation. For more information about different types of forms, please refer to the paragraph "Table Forms and Project Forms" on page 339.
- **Table Form** (also called a "standard" form): Form associated with a table. This type of form cannot be used in a component.

It should be noted that a database can be both a "matrix" and a "host," in other words, a matrix database can itself use one or more components. However, a component cannot use "sub-components" itself.

Protection of Components: Compilation

By default, all the project methods of a matrix database installed as a component are potentially visible from the host database. In particular:

- The shared project methods are found on the "Methods" page of the Explorer and can be called in the methods of the host database (see the paragraph "Sharing of Project Methods" on page 1280). Their contents can be selected and copied in the preview area of the Explorer. They can also be viewed in the debugger.

It is nevertheless not possible to open them in the Method editor nor to modify them.

- The other project methods of the matrix database do not appear in the Explorer but they too can be viewed in the debugger of the host database.

To protect the project methods of a component effectively, simply **compile** the matrix database and provide it in the form of a .4dc file (compiled database that does not contain the interpreted code). When a compiled matrix database is installed as a component:

- The shared project methods are shown on the "Methods" page of the Explorer and can be called in the methods of the host database. However, their contents will not appear in the preview area nor in the debugger.
- The other project methods of the matrix database will never appear.

Installation of a Component

To install a component in a 4D v11 database, simply copy the structure file of the matrix database into the **Components** folder of the host database. You can use aliases (Mac OS) or shortcuts (Windows).

This is handled transparently in client-server mode.

Components Folder

The Components folder must be placed next to the structure file of the host database. In the case of databases that are compiled and merged with 4D Volume Desktop, the Components folder must be next to the executable file (under Windows) and in the Contents folder of the package (under Mac OS).

4D looks for matrix databases of the **.4db** (interpreted matrix database), **.4dc** (compiled matrix database) or **.4dbase** (package type matrix database, see the paragraph ".4dbase Extension" on page 45) types in the Components folder. Other elements, in particular data files or user structure files (.4DA), are ignored.
You can use aliases or shortcuts to these matrix databases. This can be particularly useful during the development phase of a component since any changes made in the matrix database are immediately passed on to all the host databases.

In the Components folder, you can put a shortcut (Windows) or an alias (Mac OS) to another database that is itself used as a host database. By using aliases, a host database can thus become a component and vice versa. In this case, keep in mind that only one level of components is loaded. The components of a host database being used itself as a component will not be loaded.

The Components folder can contain any custom files or folders that are necessary for the operation of the component (xliff, pictures, and so on). On the other hand, it cannot contain plug-ins or Components subfolders. If either of these items are present, they will be ignored by 4D.

The plug-ins used by the components must be installed in the host database or in 4D itself (see the paragraph "Installing Plug-ins" on page 47).

Interpreted / Compiled / Unicode

A host database running in interpreted mode can use either interpreted or compiled components, in Unicode mode or not. It is possible to install both interpreted and compiled components in the same host database. However, if several compiled components are present, they must be executed in the same Unicode mode.

A host database running in compiled mode cannot use interpreted components. In this case, only compiled components can be used. Similarly, the Unicode mode must be the same for host databases and components.

The following table summarizes the component use possibilities:

		Interpreted components		**Compiled components**	
		Unicode	*Non-Unicode*	*Unicode*	*Non-Unicode*
Interpreted host database	*Unicode*	√		√ (*)	
	Non-Unicode				
Compiled host database	*Unicode*	-	-	√	-
	Non-Unicode	-	-	-	√

(*) If several compiled components are installed, they must operate in the same Unicode mode.

Notes

- An interpreted host database that contains interpreted components can be compiled if it does not call methods of the interpreted component. Otherwise, a warning dialog box appears when you attempt to launch the compilation and it will not be possible to carry out the operation.
- In general, an interpreted method can call a compiled method, but not the reverse, except via the use of the EXECUTE METHOD and EXECUTE FORMULA commands.

For more information about inter-component and host database-component exchanges, please refer to the paragraph "Interaction Between Components and Host Databases" on page 1278.

Mac OS / Windows

An interpreted component developed under Mac OS can be installed in a Windows environment and vice versa.
On the other hand, compiled components must be executed on the same type of platform they were compiled on, unless they were compiled for both platforms.

Client-Server

Components installed in the server database are automatically transferred to the client machines via a mechanism resembling that of plug-ins.

On the other hand, it is not recommended to modify a component in client/server mode since the changes will be stored locally and the component will not be updated on the server machine.

Loading of Components on Startup

The components are loaded when the host database is opened.

If a component contains compiled code and interpreted code that do not correspond, an error message is displayed and the component is not loaded in the host database.

If a component is missing on startup, the host database will open normally. This means that it is possible to set up applications using optional components. It is possible to check for the presence of components using the COMPONENT LIST command.

Naming Conflicts (Masking Methods)

Unlike other shared objects (see the paragraph "Shared and Unshared Objects" on page 1278), shared project methods have a "physical" existence in the database and are not simply created by code execution.

Consequently, naming conflicts can occur when a shared project method of the component has the same name as a project method of the host database. In this case, when the code is executed in the context of the host database, it is the host database method that is called. This means that it is possible to "mask" a component method with a custom method (for example to obtain a different functionality).
Of course, when the code is executed in the component, it is the component method that is called.

This masking will be indicated by a warning in the event of compilation of the host database.

Note If two components share methods having the same name, an error is generated when the host database is compiled.

Developing a Component

Since a component comes in the form of a 4D database, developing a component is similar to developing a functionality in a database.

There are nevertheless certain restrictions and specific rules related to the nature of components.

Usable and Unusable Objects

A component can call on most of the objects of 4D: project methods, project forms, menu bars, choice lists, pictures from the library, and so on.

Only the following objects cannot be used by a component:

- Tables and fields,
- Table forms and their associated form methods (on the other hand, a component can call a table form of the host database),
- User forms,
- Database methods and triggers.

It is not necessary to remove these elements if they exist in the matrix databases. When an "unusable" object is present, it is simply ignored once the component is installed.

Note Users and groups as well as any access rights that may have been set in the matrix database are ignored in the host database.

Only project methods that are "shared" by the component are visible and can be selected in the host database in Design mode. On the other hand, project methods that are "shared" by the host database can be called by the component. For more information about this, please refer to the paragraph "Sharing of Project Methods" on page 1280.

Other component objects (project forms, choice lists, menus and so on) can be used by the component but will not be accessible as structure objects from the host database. Note that certain objects are *partitioned* and others are *shared* between the host database and the component(s). For more information about this, please refer to the paragraph "Shared and Unshared Objects" on page 1278.

A component can use the plug-ins installed in the 4D application or in the host database. It is not possible to install plug-ins in the component folder.

The database methods and generic parameters of matrix databases (Web folder, Preferences, and so on) are never taken into account.

Unusable Commands

The following commands are not compatible for use within a component because they modify the structure file — which is open in read-only. Their execution in a component will generate the error -10511, "The *CommandName* command cannot be called from a component":

- ON EVENT CALL
- Method called on event
- SET PICTURE TO LIBRARY
- REMOVE PICTURE FROM LIBRARY
- SAVE LIST
- ARRAY TO LIST
- EDIT FORM
- CREATE USER FORM

- DELETE USER FORM
- CHANGE PASSWORD
- EDIT ACCESS
- Set group properties
- Set user properties
- DELETE USER
- CHANGE LICENSES
- BLOB TO USERS
- SET PLUGIN ACCESS

Note The Current form table command returns Nil when it is called in the context of a project form. Consequently, it cannot be used in a component.

Use of Forms

Only “project forms” (forms that are not associated with any specific table) can be used in a component. Any project forms present in the matrix database can be used by the component.

For more information about project forms, please refer to the paragraph “Table Forms and Project Forms” on page 339.

In a host database, it is not possible in Design mode to display nor to explicitly call a project form that comes from a component. These forms do not appear on the “Forms” page of the Explorer nor in the Method editor. They can only be called via the project methods of the component.

A component can call table forms of the host database. Note that in this case it is necessary to use pointers rather than table names between brackets [] to specify the forms in the code of the component.

Note If a component uses the ADD RECORD command, the current Input form of the host database will be displayed, in the context of the host database. Consequently, if the form includes variables, the component will not have access to it (see the paragraph “Interaction Between Components and Host Databases” on page 1278).

Use of Resources

Components can use resources (“conventional” Mac OS resources or XLIFF type files).

In conformity with the resource management principle (see the paragraph “Database Architecture” on page 42), the resource files of components must be placed in a Resources folder, located next to the .4db or .4dc file of the component. If the component is of the .4dbase architecture (recommended architecture), the Resources folder must be placed inside this folder.

Automatic mechanisms are operational: the XLIFF files found in the Resources folder of a component will be loaded by this component. A component will also automatically use the “conventional” Mac OS resources located in the .rsr file next to the .4db or .4dc file. “Conventional” resource files located in the Resources folder must be explicitly loaded into the component using the commands of the “Resources” theme.

In a host database containing one or more components, each component as well as the host databases has its own “resources string.” Resources are partitioned between the different databases: it is not possible to access the resources of component A from component B or the host database (see the paragraph “Shared and Unshared Objects” on page 1278).

On-line Help for Components

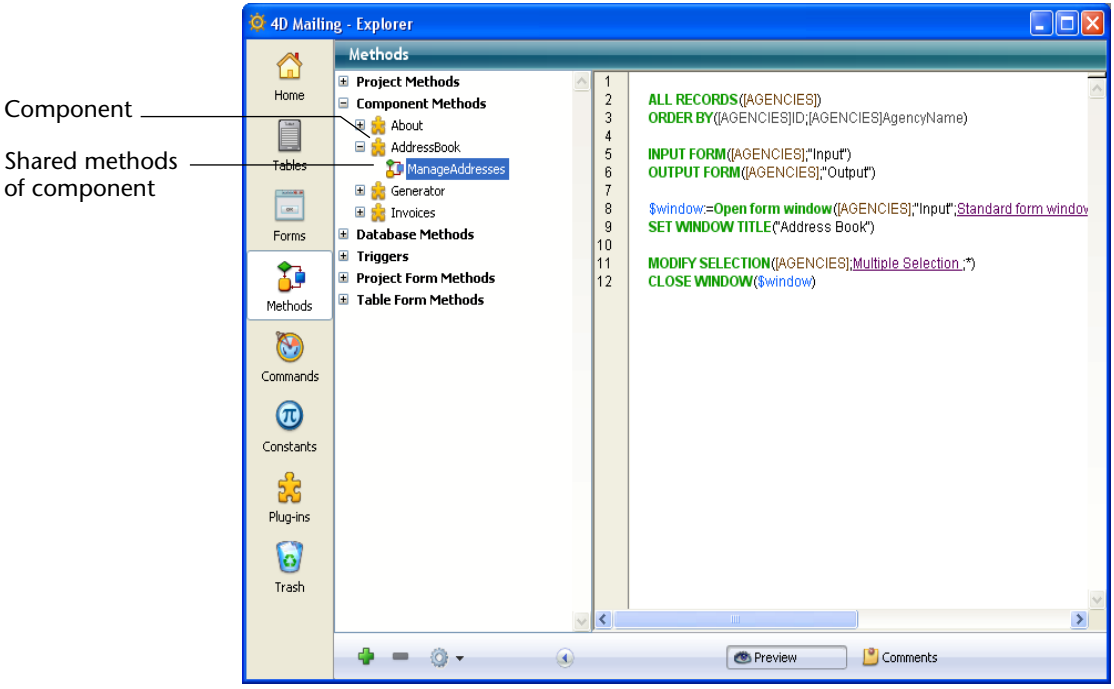
A specific mechanism has been implemented in order to allow developers to add on-line help to their components. The principle is the same as that provided for 4D databases (see appendix A on page 1285) :

- The component help must be provided as a file suffixed *.htm*, *.html* or (Windows only) *.chm,*
- The help file must be put next to the structure file of the component and have the same name as the structure file,
- This file is then automatically loaded into the **Help** menu of the application with the title “**Help for...**” followed by the name of the help file.

Interaction Between Components and Host Databases

Display of Components

When a component has been installed in a host database, its name appears on the Methods page of the Explorer of the host database, in the **Component Methods** theme. Shared project methods are listed as hierarchical lists and, if the component is interpreted, their contents can be displayed in the preview area.



Note For more information about the definition of shared methods, please refer to the paragraph “Sharing of Project Methods” on page 1280.

Shared and Unshared Objects

Certain types of objects defined by a component evolve in their own execution space, which eliminates the possibility of conflicts arising with the existing objects of the host database and those of other components. These objects are called “unshared” or “partitioned.” For example, variables are partitioned, which means that a *<>Myvar* variable of the Longint type that is created and used by a component can coexist with a *<>Myvar* variable of the Text type that is created and used by the host database (or by another component).

Other objects share the same execution space between the host database and the components. Using these objects requires more precautions to be taken but permits the host database and the components to communicate with each other. These objects are called "shared" or "non-partitioned."
For example, sets are non-partitioned, which means that if a component creates the *mySet* set, it will be deleted if the host database executes the statement CLEAR SET(mySet).

Unshared Objects

The following are **unshared** (partitioned) objects:

- Style sheets
- Help tips
- Choice lists
- Library pictures
- Menus and menu bars created via the Menu editor
- Project methods not having the "Shared by components and host database" property
- Semaphores
- Processes
- Variables (local, process and interprocess)
- System variables (OK, Document, etc.)
- Project and table forms
- Resources and references to open resources files

Shared Objects

The following are **shared** (non-partitioned) objects:

- Sets
- Named selections
- Hierarchical lists using a reference (created via the New list, Load list, Copy list or BLOB to list commands)
- Menus and menu bars using the ID returned by the Create menu command.

- Project methods with the “Shared by components and host database” property
- XML structure references
- Open file references (except for resources files)
- Pointers

Note Of course, unusable objects found in the matrix database are ignored by the host database (see the paragraph “Usable and Unusable Objects” on page 1274).

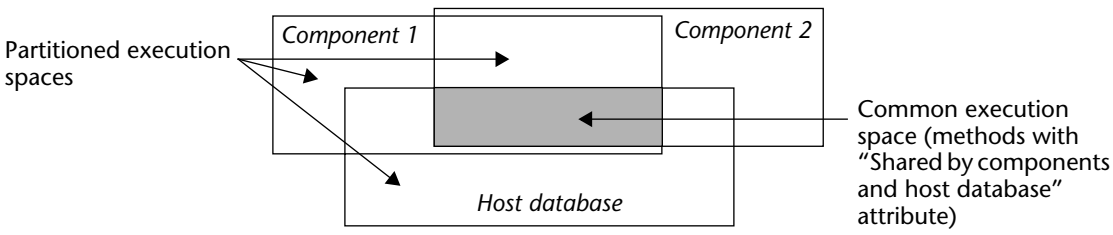
Sharing of Project Methods

All the project methods of a matrix database are by definition included in the component (the database is the component), which means that they can be called and executed by the component.

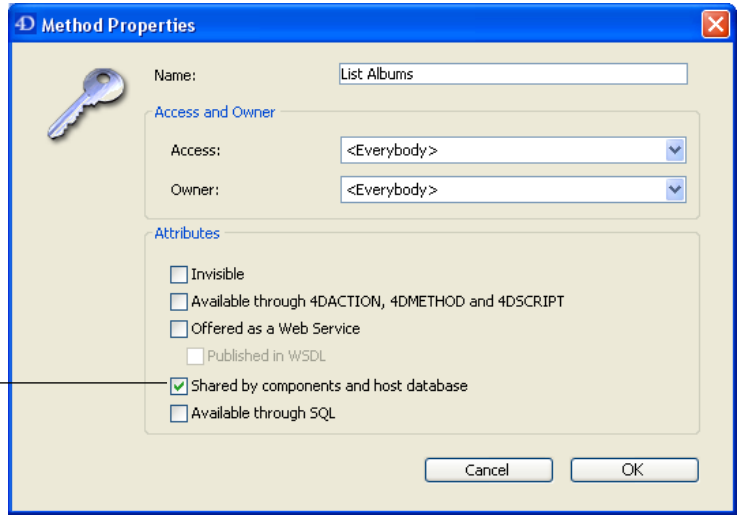
On the other hand, by default these project methods will not be visible, nor can they be called in the host database. In the matrix database, you must explicitly designate the methods that you want to share with the host database. These project methods will be visible on the Methods page of the Explorer (see the paragraph “Display of Components” on page 1278) and can be called in the code of the host database (but they cannot be modified in the Method editor of the host database). These methods form **entry points** in the component.

Conversely, for security reasons, by default a component cannot execute project methods belonging to the host database. In certain cases, you may need to allow a component to access the project methods of your host database.
To do this, you must explicitly designate the project methods of the host database that you want to make accessible to the components.

Sharing of methods

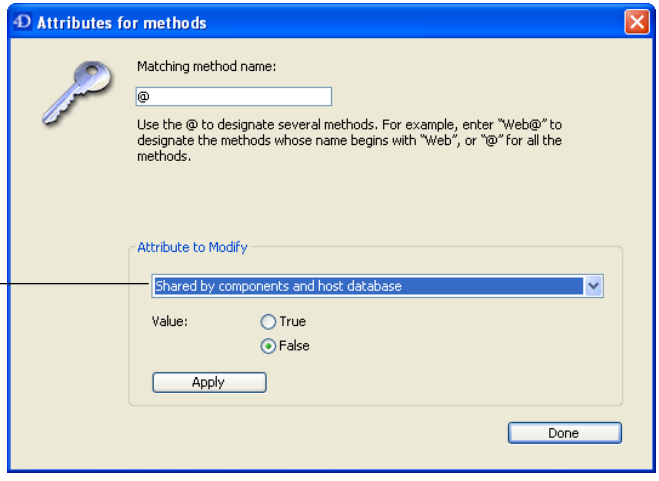


This configuration is carried out via the **Shared by components and host database** property in the Method Properties dialog box:



New option for method accessibility

It is also possible to apply this property to several different methods at once using the Batch setting of attributes dialog box (which can be accessed using the contextual menu of the Explorer):



New option for method accessibility

The effect of this option is defined by the context of the database use: if the database is used as a component, the method will be accessible in the host database and visible in the Explorer. If the database is a host database, the method will be usable by the components.

Passing Variables

The local, process and interprocess variables are not shared between components and host databases. The only way to access component variables from the host database and vice versa is using pointers.

- Example using an array:
  - In the host database:
    ARRAY INTEGER (MyArray;10)
    AMethod (->MyArray)
  - In the component, the *AMethod* project method contains:
    ADD ELEMENT($1->; 2)
- Examples using variables:

  C_TEXT(myvariable)
  component_method1(->myvariable)

  C_POINTER($p)
  $p:=*component_method2(...)*

  The use of pointers to allow components and the host database to communicate requires taking the following specificities into account:

- The Get pointer command will not return a pointer to a variable of the host database if it is called from a component and vice versa.
- The component architecture allows the coexistence, within the same interpreted database, of both interpreted and compiled components (conversely, only compiled components can be used in a compiled database).
  The use of pointers in this case must respect the following principle: the interpreter can unpoint a pointer built in compiled mode but, conversely, in compiled mode, it is not possible to unpoint a pointer built in interpreted mode.
  Let's illustrate this principle with the following example: given two components, C (compiled) and I (interpreted), installed in the same host database:
  - If component C defines the *myCvar* variable, component I can access the value of this variable by using the pointer ->*myCvar*.
  - If component I defines the *myIvar* variable, component C cannot access this variable by using the pointer ->*myIvar*. This syntax causes an execution error.

- The comparison of pointers using the RESOLVE POINTER command is not recommended with components since the principle of partitioning variables allows the coexistence of variables having the same name but with radically different contents in a component and the host database (or another component). The type of the variable can even be different in both contexts.
If the *myptr1* and *myptr2* pointers each point to a variable, the following comparison will produce an incorrect result:

```
RESOLVE POINTER(myptr1;vVarName1;vtablenum1;vfieldnum1)
RESOLVE POINTER(myptr2;vVarName2;vtablenum2;vfieldnum2)
If(vVarName1=vVarName2)
   ` This test returns True even though the variables are different
```

In this case, it is necessary to use the comparison of pointers:

```
If(myptr1=myptr2)
   ` This test returns False
```

Access to Tables of the Host Database

Although components cannot use tables, the following commands can be called within a component:

- DEFAULT TABLE
- NO DEFAULT TABLE
- Current default table

In fact, these commands are useful when a component must use tables of the host database. The pointers will permit the host database and the component to communicate with each other in this case.

For example, here is a method that could be called from a component:

```
C_LONGINT($1) `Number of a table in host database
$tablepointer:=Table($1)
DEFAULT TABLE($tablepointer->)
CREATE RECORD `Use the default table of the host database
$fieldpointer:=Field($1;1)
$fieldpointer->:="value"
SAVE RECORD
UNLOAD RECORD
```

Scope of Language Commands

Except for prohibited commands (see the paragraph "Unusable Commands" on page 1275), a component can use any command of the 4D language.

When commands are called from a component, they are executed in the context of the component, except for the EXECUTE METHOD command that uses the context of the method specified by the command. Also note that the read commands of the "Users and Groups" theme can be used from a component but will read the users and groups of the host database (a component does not have its own users and groups).

The SET DATABASE PARAMETER and Get database parameter commands are an exception: their scope is global to the database. When these commands are called from a component, they are applied to the host database.

Furthermore, specific measures have been specified for the Structure file and Get 4D folder commands when they are used in the framework of components (see the *Language Reference* manual).

Note The COMPONENT LIST command can be used to obtain the list of components that are loaded by the host database.

Debugging

When you use components that are not compiled, their code appears in the standard debugger of the host database.

The debugger respects the execution space of partitioned objects. If you display the value of the *var1* variable of the host database in the custom watch pane then execute the code belonging to the component also containing a *var1* variable, the value displayed will not be updated. You must display another instance of the variable in the custom watch pane to obtain its value in the current context.

A Assigning a Custom Help File

4D allows you to associate a custom on-line help file with each database.

The on-line help system provided with 4th Dimension is compatible with each 4D work environment (stand-alone database or client-server, interpreted or compiled, run using 4D Desktop Interpreted, or integrated into a 4D Desktop Volume, etc.).

4D supports two help files formats: CHM and HTML. These formats correspond to the industry standards.

Moreover, you can associate a specific section of this help file with each of the database's forms, which allows you to provide contextual on-line help.

Setting the Database On-line Help

File Formats

The format of the files must be one of the following:

- A ".CHM" file, which is the standard on-line help format for Windows. CHM files are opened by the *Microsoft® HTML Help Executable* (HH.EXE) application, integrated into the operating system. These files are compatible with Windows XP and Windows Vista. Microsoft® provides the *HTML Help Workshop* application free of charge, which is needed for preparing and compiling the .CHM files.
- A ".HTM" file, which is the HTML format. With this format you can create an identical on-line help on all platforms. The on-line help is then displayed in a Web browser.

Assigning the Help File to the Database

Once the Help file generated, you need to associate it with a database so that it is opened when users call the on-line help (see the "Calling the On-line Help from a Database" paragraph on page 1287). To assign a help file to a database, make sure:

- The name of the help file is identical to the name of the database's structure file. It must also have the ".CHM" or ".HTM" extension, depending on its format (see above) and platform.
- The help file is placed next to the structure file or in the Plugins folder located at the same level as the database structure file.

4D Server If you want the help file to be accessible to all the client workstations, place it in the Plugins folder. It will then be transferred to the client workstations automatically.

Note • Plug-ins can have a help file, which must be placed in the Plugins folder both in a single-user application or in a client/server. The help file of the plug-in must have the same name as the plug-in with the extension .CHM or .HTM.
• Components can also include a custom help file, placed next to the structure file of the component. For more information, please refer to the "On-line Help for Components" paragraph on page 1277.

Creating Contextual On-line Help

Creating contextual on-line help is done by associating a section number with each of your forms. When you call the on-line help from a form, the corresponding help page is displayed. When the user calls the on-line help, the help topic that has the same ID as the form is displayed.

Contextual on-line help is available:

- Under Windows, if the on-line help is in CHM format.
- Under Windows and Mac OS, if the on-line help is in HTML format.

The assignment of an ID number to a form is done in the form properties; for more information, refer to the "Contextual On-line Help" paragraph on page 409.

Once you have assigned the ID number to the form, you need to assign the same ID number to the help file. This operation varies according to the format you are using:

- For CHM files, refer to the documentation of the *HTML Help Workshop* application.
- For HTML files, you must declare each section using an anchor and assign a number.
  A section is declared by using a marker of the <A NAME="Section"> type. For example, <A NAME="21">.
  The URL of the section has the following form: <A HREF="#Section">...</A>. For example, <A HREF="#21">...</A>.

If the section number passed in the form is 0 or if it does not exist in the file, 4D displays the first page of the help file.

Calling the On-line Help from a Database

In 4D, you can call a database's custom on-line help in two ways:

- By choosing **DatabaseName Help** in the **Help** menu. In this case, the first page of the help file is displayed.
- By pressing the **F1** key (Mac OS and Windows) or the **Help** key (Mac OS only) when a form is displayed in Application mode. In this case, if a help topic number has been associated with the form, the corresponding page is displayed (contextual help); otherwise the first page of the help file is displayed.

Note Under Mac OS, the functioning of the **F1** key can be customized. In Application mode, when the user presses the **F1** key:
• If a "*DatabaseName*.HTM" file exists next to the database structure file, the standard mechanism for managing 4D on-line help files is implemented. The help file is opened in a default browser window.
• Otherwise, 4D does nothing and the developer must process the event as desired. This means it is possible to set up completely customized help systems or to assign another function to the **F1** key.

B Command Line Interface (CLI) Under Mac OS

You can use a command line in the Mac OS X Terminal to drive your 4D applications (4D Developer and 4D Server). This function allows you to open or close a database remotely, which can be particularly useful for administering Web servers.

Basic Information

To be able to execute most 4D commands using the Mac OS X Terminal, you must first access the folder where the application is found inside the package (Contents/Mac OS path).
For example, if the 4D Developer package is located in the *MyFolder* folder, you must write the command line as follows:

```
/MyFolder/4D\ Developer.app/Contents/Mac OS/4D\ Developer
```

Note You can also insert the command line using drag and drop with the application icon in the Terminal.

It is recommended that you pass the "&" character at the end of the command line so that the application is executed as a background task in a separate process. For example:

```
/MyFolder/4D\ Developer.app/Contents/Mac OS/4D\ Developer &
```

Commands and Arguments

Here is a description of command lines and their arguments which are supported by 4D applications:

- **Launch application**

  Syntax: Path/ApplicationName

  Example:
  4D\ Developer.app/Contents/Mac OS/4D\ Developer &

  The action of this command is identical to double-clicking the 4D Developer application. The application is executed and the select database dialog box appears.

- **Launch application with a structure file**

  Syntax: Path/ApplicationName StructurePath

  Example:
  4D\ Developer.app/Contents/Mac OS/4D\ Developer /Users/Roger /Databases/MyDatabase/MyDatabase.4DB

  This command launches the application and opens the designated structure file with the current data file. No dialog box appears.

- **Launch application with a structure file and a data file**

  Syntax: Path/ApplicationName -d DataPath StructurePath

  Example:
  4D\ Developer.app/Contents/Mac OS/4D\ Developer -d /Users/Roger/Databases/MyDatabase/MyDatabase.4DD /Users/Roger/Databases/MyDatabase/MyDatabase.4DB

  This command launches the application and opens the designated structure file with the selected data file. No dialog box appears.

- **Quit application**

  Syntax: kill -s INT process_ID

  Example:
  Kill -s INT 323

  The action of this command is identical to selecting the **Quit** command. The processes are shut down one by one; the cache is saved; and the application quits.

Note The process number of the 4D application (*process_ID*) can be obtained using commands such as **ps** or **top**.

- **Help**

  Syntax: Path/ApplicationName -h

  Example:
  4D\ Developer.app/Contents/Mac OS/4D\ Developer -h

  This command displays a summary of commands that can be used with 4D in the Mac OS X Terminal.

- **Launch the application in a user session other than the current session**

  Syntax: Path/ApplicationName -u UserName -g GroupName

  Example:
  4D\ Developer.app/Contents/Mac OS/4D\ Developer -u john -g accnt

  This command forces the application to launch using the specified user session (the current user session is used by default).

C XLIFF Architecture

4D supports the XLIFF standard for the localization of interface text and titles. This technology is used internally for 4D applications, and 4D developers as well as plug-ins developers will be able to benefit from this new implementation in their own customized applications and plug-ins.

The principle for setting up a 4D application interface that is translated dynamically is as follows: all the elements that depend on the language (text, labels and pictures) are stored outside the application, as files, in a folder named *Resources*. In forms, menus, and so on, the labels and pictures are specified as references. When the application is executed, these elements are displayed dynamically from external files depending on the linguistic context. The XLIFF standard uniformizes the mode of referencing and displaying labels.

Compatibility Note In addition to XLIFF, 4D still supports the previous system, based on the concept of "resources" (of the STR# type). This system is nevertheless now obsolete. Both systems can be used at the same time, in particular in converted applications (this point is detailed in the following pages).

What is XLIFF?

XLIFF (*XML Localization Interchange File Format*) is a dedicated standard for the localization process. It is used to describe the correspondence between a source language and a target language within an XML file.

Actually, the XLIFF standard is a new alternative to resources-based localization systems. Various tools, including several freeware, can also be found to manage such files.

For more information about the XLIFF standard, please refer to the official XLIFF 1.1 Specification that can be found at the following address:

http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm

Warning: The XLIFF standard is case sensitive.

Calling XLIFF Strings from a 4D Database

Location of References

In 4D, XLIFF can be used at the following locations:

- "Object Name" field in the Property List of the Form editor
- "Window Title" field in the Form Properties dialog box that can be accessed from the Explorer
- Static text, including titles of buttons, checkboxes, etc.
- Menu editor (except *MyTitle* syntax, see below)
- Help Tips editor (except *MyTitle* syntax, see below).

Notes

• In order for the XLIFF reference to be recognized, it must be located at the beginning of the text area.

• It is not possible to display XLIFF references in the Help Tips editor. To use help tips based on XLIFF, enter the XLIFF references directly in the "Help Tip" field of the Property List.

Syntax of References

In 4D, you can use XLIFF references via one of the following syntaxes:

- **1015,3** syntax
  When this syntax is used with XLIFF files, the first value (1015 in the example) designates the **id** attribute of the **group** element.
  The second value (3 in the example) designates the **id** attribute of the **trans-unit** element.

Compatibility Note This syntax is exactly the same as that of the "former generation" STR# resources, of the type ":xxxx,yyy" where xxxx is the number of the STR# resource and yyy is the element number. For example, ":10115,3" means that 4D must use the 3rd element of the STR# resource number 10115.
This way you can keep a resource-based system in a converted database while installing a new XLIFF system. In fact, 4D will first try to locate and use the values corresponding to 1015,3 in all loaded XLIFF files; then, if it does not find the string, it will search for it in all the open resource files. With this mechanism, you can implement an XLIFF-based system in an application without having to edit your existing dynamic string references: you just need to copy an XLIFF file at the correct location (see below) and it will be taken into account by 4D. Both systems can be used simultaneously. In all cases, loaded XLIFF files have priority over resource files if the same string is present in both locations.

- **xliff:OKButton** syntax
  This alternative syntax can only be used with loaded XLIFF files. In this case, the referenced value (*OKButton* in the example) designates the **resname** attribute of the **trans-unit** element.

- **MyTitle** syntax (in other words, static text or titles "as written")
  The purpose of this syntax is to facilitate the translation of forms — unlike the other syntaxes, it is only used with forms. The principle is to surround the **trans-unit** elements in the XLIFF file by two **group** elements whose **resname** attributes contain, respectively, the table name and the form name.
  - Example (*table form*)
    To translate the title of the "Save" button ("Title" field in the Property list) in the "Form1" form of the [Clients] table, the XLIFF file simply need to contain the lines:

```
<group resname="[Clients]">
  <group resname="Form1">
    <trans-unit resname="Save">
      ...
    </trans-unit>
  </group>
</group>
```

- Example (*project form*)
  In a project form, the table name must be replaced by [Project-Form], which produces the following:

```
<group resname="[ProjectForm]">
  <group resname="Form1">
    <trans-unit resname="Save">
      ...
    </trans-unit>
  </group>
</group>
```

If you use different syntaxes in your database, the order of priority applied when searching a valid translation in the XLIFF files will be as follows:
1. “:1015,3” syntax
2. “:xliff:OKButton” syntax
3. “MyTitle” syntax.

Certain commands of the “Resources” theme have been modified in order to take advantage of XLIFF documents. For more information, please refer to the *Language Reference* manual.

It is possible to view the XLIFF reference or the title translated according to the current language in the form editor using, respectively, the **Show Name** and **Show Resource** commands of the **Object** menu of the form editor. For more information, please refer to the paragraph “The Object Menu” on page 385.

Installing Customized XLIFF Files

To implement an XLIFF architecture for your localized application, you just need to build one or more valid XLIFF file(s) and copy them into the *Resources* folder of the database.

Note For more information about the *Resources* folder of the database, please refer to the paragraph “Database Architecture” on page 42.

Here is the full pathname to use:

Windows:
***MyDatabase*\Resources*Lang*.lproj*MyLocEn*.xlf**

Mac OS:
***MyPackage*:Resources:*Lang*.lproj:*MyLocEn*.xlf**

where:

- *MyDatabase* is the folder that contains the database files and *MyPackage* is the database package under Mac OS.
- *Lang*.lproj is a folder that contains the XLIFF files for the *Lang* language. The name of the folder must comply with the international standard (see the following paragraph). For example, for English versions the folder should be named *en.lproj*.

4D will automatically load the XLIFF files in the folder corresponding to the current language of the database. To set the current language of the database, 4D carries out successive searches in the *Resources* folder of the database to find a language corresponding to (in the following order of priority):
1. The system language (under Mac OS, several languages can be set by order of preference, 4D uses this setting).
2. The language of the 4D application.
3. English.
4. If none of these searches find anything, the first language found in the *Resources* folder is loaded.

If a language variation is used and is not available in the XLIFF files, the next closest language will be used.

XLIFF files can be named freely; they just need to have the extension ".xlf.". You can place several XLIFF files in the same language folder; they will be loaded in the alphabetical order of the file names.

.lproj Folder Name

The name of the ".lproj" folder must respect one of the standards described below. 4D will look for a valid folder name based on each of these standards, in the following specific order.

1 **RFC 3066 bis**

Currently, this RFC is still at the draft stage. 4D supports the following part of this RFC: A language is described by a Country code (ISO639-1) + a hyphen + a Regional code (ISO3166).
For example, "fr-ca" (thus *fr-ca.lproj*) for the French-Canadian language.

2 **ISO639-1**

This standard defines each language by two letters. For example, "en" (thus en.*lproj*) for the English language.
Ref: http://www.loc.gov/standards/iso639-2/php/English_list.php

3 **Legacy name**

In this convention, the language name is written out completely in English. For example, "english" (thus *english.lproj*) for the English language.

Note The first two conventions are supported only for Mac OS version 10.4 or higher. With previous versions of this OS, only the "Legacy" name can be used.

A summary table of the language codes supported by 4D is provided in the paragraph "Language Codes" on page 1299.

If several language definitions are found, 4D will always look for the most precise translations. For example, if the current OS language setting is "French Canadian," 4D will first look for "fr-ca" translations and then, if not found, "fr" translations.

Note The same principle applies within XLIFF files for the "target-language" tag. You must be sure to carefully set this attribute within the XLIFF files since a file located in the "fr-ca.lproj" folder that has a "target-language=fr" tag will be considered as a translation into "fr" and not "fr-ca."

Reloading Customized XLIFF Files

The XLIFF files are reloaded “dynamically” while you localize your application, so that you can see if the localized words or sentences fit into the object frames (buttons, group boxes, etc.). This reload occurs as 4D comes back to the front whenever the modification date or time has been changed since the last load.The current form will be reloaded simultaneously.

Language Codes

The following table lists the language codes supported by 4D for managing XLIFF files.

Languages	ISO639-1	“Legacy”	ISO3166
AFRIKAANS	af	afrikaans	
ALBANIAN	sq	albanian	
ARABIC_SAUDI_ARABIA	ar	arabic	sa
ARABIC_IRAQ	ar	arabic	iq
ARABIC_EGYPT	ar	arabic	eg
ARABIC_LIBYA	ar	arabic	ly
ARABIC_ALGERIA	ar	arabic	dz
ARABIC_MOROCCO	ar	arabic	ma
ARABIC_TUNISIA	ar	arabic	tn
ARABIC_OMAN	ar	arabic	om
ARABIC_YEMEN	ar	arabic	ye
ARABIC_SYRIA	ar	arabic	sy
ARABIC_JORDAN	ar	arabic	jo
ARABIC_LEBANON	ar	arabic	lb
ARABIC_KUWAIT	ar	arabic	kw
ARABIC_UAE	ar	arabic	ae
ARABIC_BAHRAIN	ar	arabic	bh
ARABIC_QATAR	ar	arabic	qa
BASQUE	eu	basque	
BELARUSIAN	be	belarusian	
BULGARIAN	bg	bulgarian	
CATALAN	ca	catalan	
CHINESE_TRADITIONAL	zh	chinese	cht
CHINESE_SIMPLIFIED	zh	chinese	chs

Languages	ISO639-1	"Legacy"	ISO3166
CHINESE_HONGKONG	zh	chinese	hk
CHINESE_SINGAPORE	zh	chinese	sg
CROATIAN	hr	croatian	
CZECH	cs	czech	
DANISH	da	danish	
DUTCH	nl	dutch	
DUTCH_BELGIAN	nl	dutch	be
ENGLISH_US	en	english	us
ENGLISH_UK	en	english	gb
ENGLISH_AUSTRALIA	en	english	au
ENGLISH_CANADA	en	english	ca
ENGLISH_NEWZEALAND	en	english	nz
ENGLISH_EIRE	en	english	ie
ENGLISH_SOUTH_AFRICA	en	english	za
ENGLISH_JAMAICA	en	english	jm
ENGLISH_CARIBBEAN	en	english	cb
ENGLISH_BELIZE	en	english	bz
ENGLISH_TRINIDAD	en	english	tt
ESTONIAN	et	estonian	
FAEROESE	fo	faorese	
FARSI	fa	persian	
FINNISH	fi	finnish	
FRENCH	fr	french	
FRENCH_BELGIAN	fr	french	be
FRENCH_CANADIAN	fr	french	ca
FRENCH_SWISS	fr	french	ch
FRENCH_LUXEMBOURG	fr	french	lu
GERMAN	de	german	
GERMAN_SWISS	de	german	ch
GERMAN_AUSTRIAN	de	german	at
GERMAN_LUXEMBOURG	de	german	lu
GERMAN_LIECHTENSTEIN	de	german	li
GREEK	el	greek	
HEBREW	he	hebrew	

Languages	ISO639-1	"Legacy"	ISO3166
HUNGARIAN	hu	hungarian	
ICELANDIC	is	iceland	
INDONESIAN	id	indonesian	
ITALIAN	it	italian	
ITALIAN_SWISS	it	italian	ch
JAPANESE	ja	japanese	
KOREAN_WANSUNG	ko	korean	
KOREAN_JOHAB	ko	korean	
LATVIAN	lv	latvian	
LITHUANIAN	lt	lithuanian	
NORWEGIAN	no	norwegian	
NORWEGIAN_NYNORSK	nn	nynorsk	no
POLISH	pl	polish	
PORTUGUESE	pt	portuguese	
PORTUGUESE_BRAZILIAN	pt	portuguese	br
ROMANIAN	ro	romanian	
RUSSIAN	ru	russian	
SERBIAN_LATIN	sr	serbian	latn
SERBIAN_CYRILLIC	sr	serbian	cyrl
SLOVAK	sk	slovak	
SLOVENIAN	sl	slovenian	
SPANISH_CASTILLAN	es	spanish	
SPANISH_MEXICAN	es	spanish	mx
SPANISH_MODERN	es	spanish	
SPANISH_GUATEMALA	es	spanish	gt
SPANISH_COSTA_RICA	es	spanish	cr
SPANISH_PANAMA	es	spanish	pa
SPANISH_DOMINICAN_REPUBLIC	es	spanish	do
SPANISH_VENEZUELA	es	spanish	ve
SPANISH_COLOMBIA	es	spanish	co
SPANISH_PERU	es	spanish	pe
SPANISH_ARGENTINA	es	spanish	ar
SPANISH_ECUADOR	es	spanish	ec
SPANISH_CHILE	es	spanish	cl

Languages	ISO639-1	"Legacy"	ISO3166
SPANISH_URUGUAY	es	spanish	uy
SPANISH_PARAGUAY	es	spanish	py
SPANISH_BOLIVIA	es	spanish	bo
SPANISH_EL_SALVADOR	es	spanish	sv
SPANISH_HONDURAS	es	spanish	hn
SPANISH_NICARAGUA	es	spanish	ni
SPANISH_PUERTO_RICO	es	spanish	pr
SWEDISH	sv	swedish	
SWEDISH_FINLAND	sv	swedish	fi
THAI	th	thai	
TURKISH	tr	turkish	
UKRAINIAN	uk	ukrainian	
VIETNAMESE	vi	vietnamese	

D Use of Specialized Dictionaries

4D includes a mechanism which can be used to set custom lists of words that will be accepted by the spell-check. This makes it possible to use specialized dictionaries containing terms specific to a given profession, field of application, company, etc., within an application.

It is possible to use several specialized dictionaries for each main language.

This principle works with 4D and 4D Write.

Note For more information about the spell-check in 4D, please refer to the paragraph "Using the Spell-check" on page 515.

Installation

To add a specialized dictionary in a given language, simply place a text format file in the subfolder of the main language, inside the *Spellcheck* folder. The file can be named freely but it must end with the ".txt" suffix (for example, "astronomy.txt").

The language subfolders are created in 4D at the following location:

- Under Windows: *(4D Application folder)*\4D Extensions\Spellcheck\
- Under Mac OS: (*4D Package)*:Contents:4D Extensions:Spellcheck:

The names of the subfolders correspond to the main languages; they must not be modified:

- English
- French
- German
- Spanish
- Norsk

Each folder contains a word file by default. You can add words to this file or add other files.

Client/Server

The spell-check files are stored in the 4D Client application. If you want to install specialized dictionaries within a client/server solution, you must make sure that the dictionaries are installed on each client machine (at the location described above).

Content of the Files

The files of the specialized dictionaries are lists of words separated by carriage returns. For example:

```
4D
Server
Desktop
```

You can add single words (e.g.: *boogie*) and compound hyphenated words (e.g.: *boogie-woogie*), but you cannot add sequences of words separated by spaces (e.g.: *Mark Smith*).

The internal format of text files differs between Windows and Mac OS, for reasons linked more particularly to character sets. You must thus create and provide a specialized dictionary file for each platform. The file must be saved only in the text format of the intended platform.

Loading and Use

Specialized dictionaries are loaded on application startup, just like standard dictionaries. The current language of the dictionaries is based on that of the application.

During a session, you can change the current dictionary using the SET DICTIONARY command. In this case, if a specialized dictionary exists in the corresponding language folder, it will be loaded.

During use, there is no difference between the processing of words from standard dictionaries and those from specialized dictionaries.

Index

Symbols/Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z