OLE Tools

Introduction

Overview of the OLE Services

What are OLE Services?

The Windows version of 4D Dimension allows you to work with OLE (Object Linking Embedding) objects in your layouts. In addition, OLE objects can be automatically saved and loaded from BLOB or picture fields.

Note: OLE services for 4D are only available under Windows.

An OLE object can contain any type of data created and handled by Windows OLE applications such as the Windows Paint desk accessory. In addition to the regular 4D data types (such as Alphanumeric, Numeric and so on), OLE services for 4D let you store and manage any data type created in Windows, including multimedia.

If you use a Windows word processor or spreadsheet that implements OLE services, you will be able to directly store documents from these applications in your 4D databases. For example, with the powerful database features of 4D, you will able to create customized archiving systems for storing, retrieving, and tracking your documents.

How to access OLE services in 4D ?

You access OLE services in 4D through OLE areas. The size and location of these areas depend on the use you want to make of them.

Creating an OLE area

OLE areas can be created in two environments:

- In external windows, dedicated to using an OLE area. This possibility is described in the **Creating an OLE Area in an External Window** section.
- In a 4D form. In this case, the area is included in fields, variables, other external areas, etc., that you have inserted into your form. This possibility is described in the **Creating an OLE Area in a Form** section.

Using OLE areas

Once the OLE areas have been created, you can use OLE services in Application or Design mode. There are two possibilities for doing this:

- Using the area "manually" via a context menu. This menu appears when you click on the OLE area with the right-hand button of the mouse. These functions are described in the **Using an OLE Area** chapter.
- Using one of four programming commands provided by the OLE Tools plug-in. These commands let you automate certain OLE functions. They are described in the **OLE_Tools Commands** chapter.

Installing OLE Tools

Starting from version 6.8 of 4D, the OLE Tools plug-in is included in 4D applications.

Deployment or compilation of databases using OLE Tools objects does not require any specific installation or setting.

Using an OLE Area

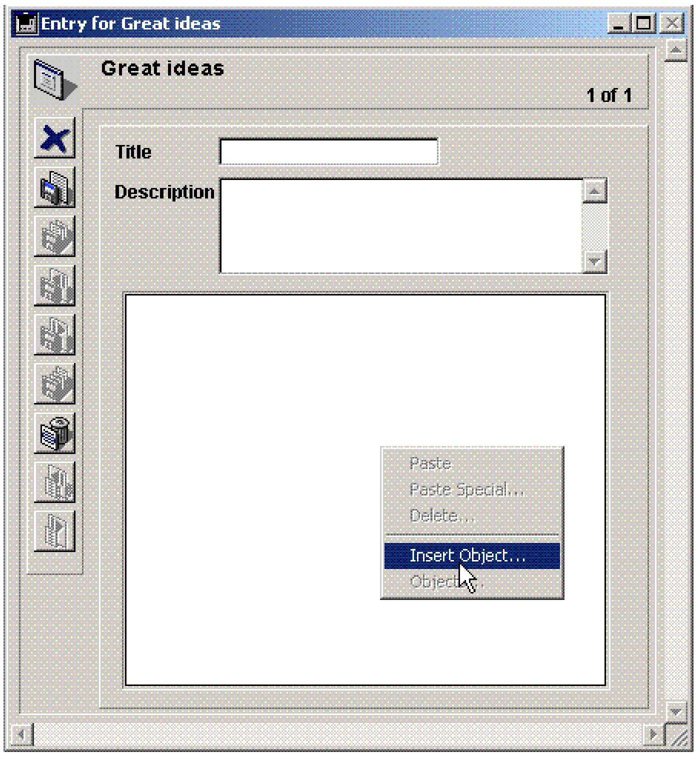
Inserting an Object

Display the OLE area in your application. You may need to execute the form.

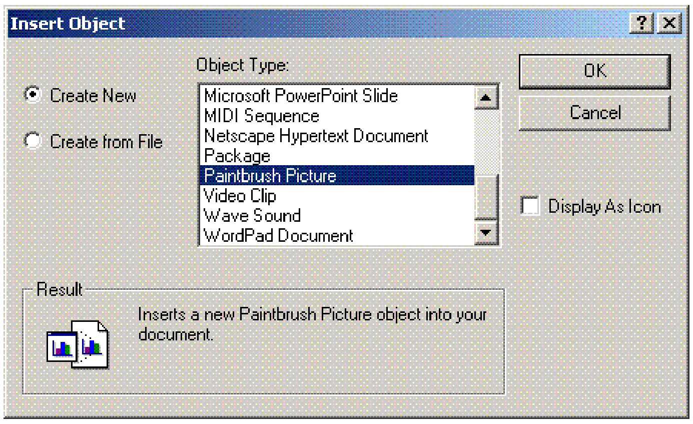
Creating and Inserting an OLE Object

If your area is located for example in a form, add a new record. Click within the OLE area whose frame should appear.

Click in the area using the **right mouse button**.
The OLE pop-up menu appears:



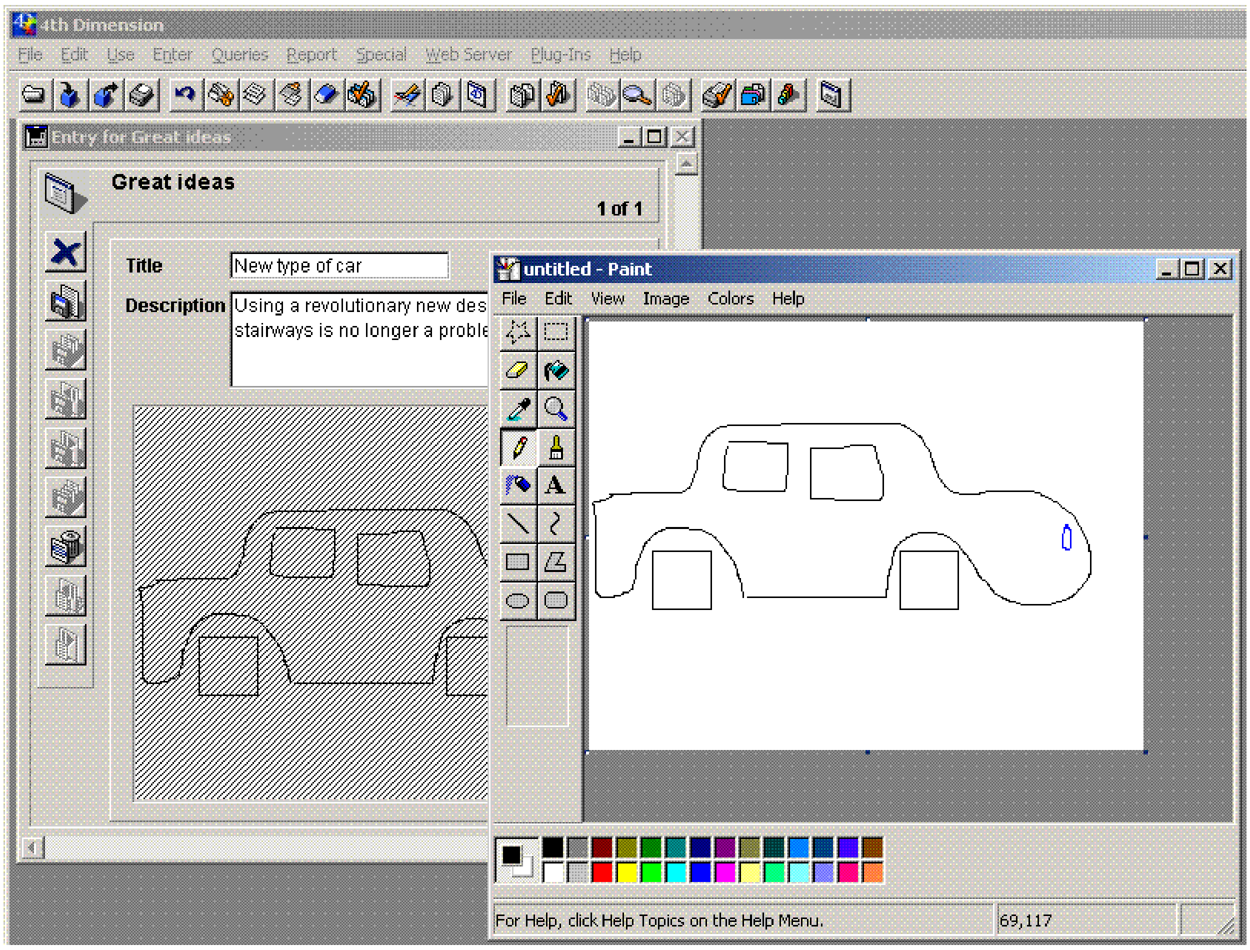
Choose **Insert Object** from the menu. The standard Insert Object dialog box is displayed. Choose for example **PaintBrush picture** and click **OK**:



Inserts a new Paintbrush Picture object into your document.

The PaintBrush Windows application is launched and opens an Untitled document. Move the PaintBrush window so you can see both this window and the OLE area in your layout.
Note that the OLE area is now displayed with cross-hatches signaling that its contents is being edited by the OLE source application.

Make a simple drawing in PaintBrush. Note that your drawings appear concurrently within PaintBrush and your 4D OLE area:


4th Dimension
File Edit Use Enter Queries Report Special Web Server Plug-Ins Help
Entry for Great ideas
Great ideas
1 of 1
Title
New type of car
Description
Using a revolutionary new des
stairways is no longer a proble
untitled - Paint
File Edit View Image Colors Help
For Help, click Help Topics on the Help Menu.
69,117

Note: With certain old versions of Windows OLE applications, you may find that the source OLE application does not update the 4D OLE client area automatically. If this happens, choose **Update** from the **File** menu of the OLE source application to force the update.

Inserting a Document into an OLE Area

In the previous example, you have created a linked and embedded object from scratch. You can also store an existing document. To do so, choose the option **Create from file** in the standard Insert Object dialog box rather than the option **Create New**.

For more information about the options of the standard OLE Insert Object dialog box, please refer to your Windows documentation.

Pasting an object via the clipboard

You can cut or copy an object in an OLE application and then paste it into an OLE area in 4D.

To do this, you just need to select the object to be inserted in the OLE application and then choose **Copy** or **Cut** in the **Edit** menu of the application.
Then, in 4D, place the mouse in the OLE area and click. Your selection will then appear in the OLE area.

Drag and drop

Instead of using the OLE pop-up menu to insert an object, you can directly drag and drop the object from an OLE source application to your client OLE area. To do so, select the data from your OLE source application, press the **Ctrl** key, click, and drag and drop the selection over your 4D OLE area.

Modifying an Object

When you are done with the insertion of your OLE object, close the source application (i. e. Paintbrush in our example).

Your object is now part of the record and can be saved on disk into your data file. Validate the data entry for this record, then reopen it: your object are loaded from the data file.

If you want to edit again, you only need to double-click on the OLE area. This will automatically launch the source application associated with your OLE object. You could also pull down the OLE pop-up menu (using the **right mouse button**) and choose **Edit** in the hierarchical **PaintBrush Picture Object** menu.

Converting an object

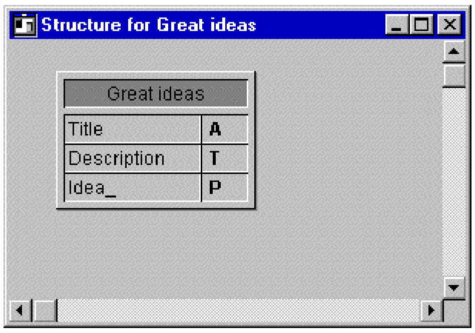
All OLE objects can be converted. The menu that appears when you click with the right-hand button of the mouse on the OLE area provides a **Convert** option. This option can be used to specify, for example, if the object will be displayed as an icon, or if such an object must be converted into an object having the format of another application.

Let's imagine a case where a friend presents you with a database that incorporates objects coming from "X" type word-processing. If you have "Y" type word-processing, when the object is opened, OLE will convert it on the fly to "Y" format and store it again in "X" format.
These operations may be penalizing, in particular if you want to make use of "Y" word-processing functions that are unknown to "X" word-processing and hence ignored during the conversion. Using the convert option, you could convert the document to "Y" format once and for all, which will simplify its use by your word-processing software and guarantee that the document will not lose any enhancements.

Creating an OLE Area

Creating an OLE Area in a Form

This section presents an example of a database that stores an OLE object. Consider the database whose structure is shown:


Structure for Great ideas

Great ideas	
Title	A
Description	T
Idea_	P

1. Create a new database and add this table to the database.
Note the presence of the picture field [Great Ideas]Idea_. The underscore at the end of the field name signals to 4D that the field can store a plug-in object, like an OLE object

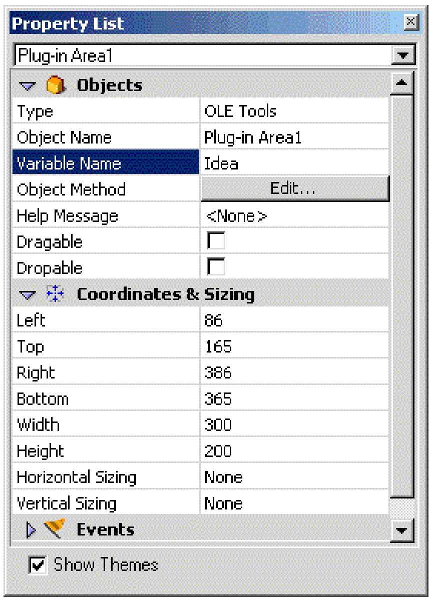
Note: You can also use a BLOB field.

Let's add an OLE object in the input form for this file.

2. In the Design environment, open the layout and choose **Insert OLE Object** from the **Object** menu.
4D inserts an OLE area as shown:

The message in the OLE object shows that 4D automatically detects the presence of the [Great Ideas]Idea_ picture or BLOB field and uses it as the container for the OLE area.

3. Double-click on the area to display the Property List:


Property List

Plug-in Area1

Objects	
Type	OLE Tools
Object Name	Plug-in Area1
Variable Name	Idea
Object Method	Edit...
Help Message	<None>
Dragable	☐
Dropable	☐
Coordinates & Sizing	
Left	86
Top	165
Right	386
Bottom	365
Width	300
Height	200
Horizontal Sizing	None
Vertical Sizing	None
Events	

☑ Show Themes

With this palette you can change the name of the area, for example, if you want to associate the area with another field in the table. To do so, you need only respect the following convention: if your area is named Name, it can be automatically stored and loaded from a picture or BLOB field named Name_ (same name followed by an underscore) that belongs to the form's table.

Creating an OLE Area in an External Window

It is possible to create an OLE Area in an independant external window. Then, the external window is dedicated to the use of the OLE area. You may:

- create the window using the 4D command **Open external window**,
- use the command **OLE tools** from the menu **Tools** of 4D.

Using the 4D command Open external window

The 4D command **Open external window** allows you to open programmatically an external area including an OLE area.

For example, the following statement creates a type 8 external window named "OLE Window" which include an OLE area:

```
myWindow:=Open external window(50;50;350;450;8;"OLE Window";"_OLE tools")
... ` Do something here
CLOSE WINDOW(myWindow)
```

This statement can be associated for example to a menu command or a button placed in a form. Once the method is executed, the variable myWindow contains the reference number of the external window. This reference number can be passed as parameter to the OLE_Tools plug-in commands in order to perform automatic actions in the window.
For information about these commands, refer to the section **Introducing the OLE_Tools commands**.

All the operations allowed by the OLE Services can also be performed manually. Refer to the chapter **Using an OLE Area**.

Using the Tools menu

You can open an external window dedicated to the use of OLE from the 4D Design mode:

1. Go to 4D Design mode.
2. Select the command **OLE tools** from the **Tools** menu.
A blank window appears, called "Untitled1". It contains an OLE Area.

You can perform in this window any action allowed by the OLE Services. For more information, refer to the chapter **Using an OLE Area**.

Note: This method can be used in the Design mode only. Indeed, as no window reference number is returned, it is not possible to manipulate the OLE area through the language commands.

OLE_Tools Commands

Introducing the OLE_Tools commands

The five OLE_Tools commands allows you to perform programmatically the following tasks:

- Inserting a file in the OLE area: *OLE_Insert file*
- Displaying the Insert Object dialog box: *OLE_INS DIALOG*
- Deleting the current object from the OLE area: *OLE_DEL OBJECT*
- Executing any command available on the OLE pop-up menu: *OLE_EXEC ACTION*
- Getting the actual size of an OLE object: *OLE_GET OBJECT RECT*

oleArea parameter

All these commands accept a Longint type variable entitled *oleArea* as first parameter. This parameter represents the internal ID of the OLE area, which is indispensable in order for the program to know to which area the processing is to be applied.

You can obtain this value in several ways, depending on the location of the OLE area and the way it was created:

- If you want to reference an OLE area inserted into a form, in the *oleArea* parameter, you must pass the name or value of the variable associated with the area. This name is specified in the property list for the objects in the 4D form editor:

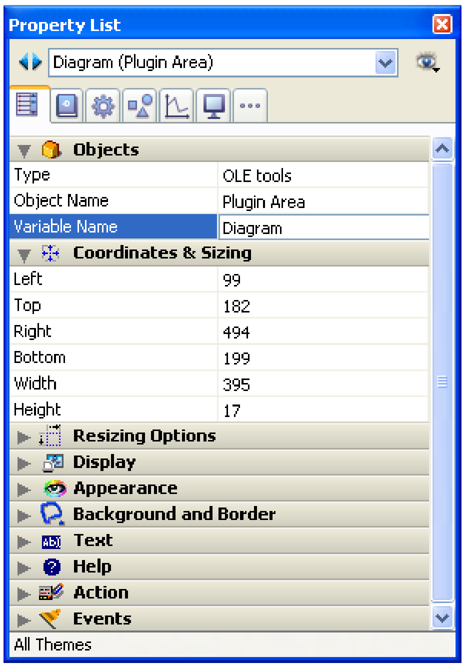

Property List

Diagram (Plugin Area)

Objects

Type	OLE tools
Object Name	Plugin Area
Variable Name	Diagram

Coordinates & Sizing

Left	99
Top	182
Right	494
Bottom	199
Width	395
Height	17

Resizing Options
Display
Appearance
Background and Border
Text
Help
Action
Events

All Themes

  In this example, the area is entitled "Diagram".

- If you want to use an OLE area placed in an external window created using the **Open external window** command, in the *oleArea* parameter, you must pass the name of the variable returned by this command. For example, if the OLE area was created using the following statement:

```
oleWindow:=Open external window(50;50;350;450;8;"Array";"_OLE tools")
```

  you must pass *oleWindow* or its value in the *oleArea* parameter.

OLE_DEL OBJECT

OLE_DEL OBJECT (oleArea)

Parameter	Type		Description
oleArea	Longint	⇒	OLE Area Reference number

Description

The *OLE_DEL OBJECT* command deletes any reference to an OLE object for the OLE area specified by *oleArea*. After a call to *OLE_DEL OBJECT*, the OLE area becomes empty. This command is equivalent to the **Delete...** menu item of the OLE pop-up menu.

OLE_EXEC ACTION

OLE_EXEC ACTION (oleArea ; action)

Parameter	Type		Description
oleArea	Longint	⇒	OLE Area Reference number
action	Integer	⇒	Number of the action in hierarchical menu

Description

The *OLE_EXEC ACTION* command executes the action specified by *action* for the OLE area specified by *oleArea*. The number you pass in *action* is the number of the menu item in the hierarchical menu that corresponds to the commands provided by the OLE source application.

Note: No matter the nature of the embedded object, the command *OLE_EXEC ACTION* does not allow the execution of the **Convert** command provided by the OLE source application.

Example

If a WAV file is inserted in an OLE area named XOleArea, the commands provided the OLE source application for WAV files are: **Play**, **Edit** and **Open**. The line:

```
OLE_EXEC ACTION(XOleArea;1)
```

plays the embedded sound file.

OLE_GET OBJECT RECT

OLE_GET OBJECT RECT (oleArea ; objectWidth ; objectHeight)

Parameter	Type		Description
oleArea	Longint	→	OLE Area Reference number
objectWidth	Longint	←	Width (in pixels) of the OLE object
objectHeight	Longint	←	Height (in pixels) of the OLE object

Description

The *OLE_GET OBJECT RECT* command returns the OLE object dimensions present in the *oleArea* area to the *objectWidth* and *objectHeight* parameters. If the OLE area contains no object then the command returns 0 for each parameter.

This command allows you to adjust the size on an OLE area included in a form according to the size of the object that it contains.

Example

In a form, you have an OLE area entitled “Sketch”. The form contains an **Adjust** button which allows you to modify the size of the area to the size of the OLE object present in the area. The method object of the button is as follows:

```
OLE_GET OBJECT RECT(Sketch;$vWidth;$vHeight)
GET OBJECT RECT(Sketch;$vLeft;$vTop;$vRight;$vBottom)
If($vWidth#0)&($vHeight#0)
   MOVE OBJECT(Sketch;$vLeft;$vTop;$vRight+$vWidth;$vBottom+$vHeight)
End if
```

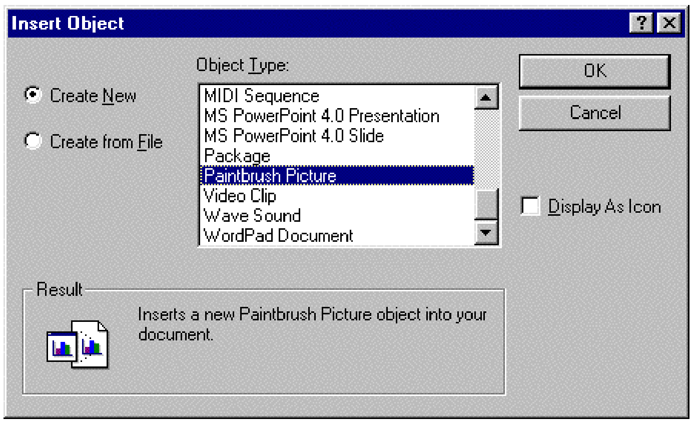
OLE_INS DIALOG

OLE_INS DIALOG (oleArea)

Parameter	Type		Description
oleArea	Longint	⇒	OLE Area Reference number

Description

The *OLE_INS DIALOG* command displays the standard Insert Object dialog box for the OLE area specified by *oleArea*. This command is equivalent to the **Insert Object...** menu item of the OLE pop-up menu.



Insert Object

Object Type:

(● Create New) ○ Create from File

MIDI Sequence
MS PowerPoint 4.0 Presentation
MS PowerPoint 4.0 Slide
Package
Paintbrush Picture
Video Clip
Wave Sound
WordPad Document

☐ Display As Icon

Result

Inserts a new Paintbrush Picture object into your document.

OLE_Insert file

OLE_Insert file (oleArea ; fileName) -> Function result

Parameter	Type		Description
oleArea	Longint	⇒	OLE Area Reference number
fileName	String	⇒	Name of the file to insert
Function result	Integer	⮌	Error code

Description

The *OLE_Insert file* command inserts the document *fileName* into the OLE area specified by *oleArea*. This command is equivalent to choosing the **Create from file** option from the standard Insert Object dialog box.

You can pass in *fileName* the "short" name of the document (if it is located in the database directory), or the full pathname to the document.

If the insertion is correctly performed, the function returns 0 (zero.) Otherwise it returns a Windows operating system error.

Example

The following method inserts a Microsoft Excel document into the OLE area Idea:

```
$DocRef :=Open document("";"XLS")
If(OK=1)
   CLOSE DOCUMENT($DocRef)
   $errCode :=OLE_Insert file(Idea;Document)
   If($errCode#0)
      ALERT("OLE error #"+String($errCode))
   End if
End if
```