# デザインリファレンス


はじめに
4Dファイルの管理
環境設定
データベース設定
エクスプローラー
ランタイムエクスプローラー
デザインモードの検索と置換
オブジェクトのドラッグ＆ドロップ
データベースストラクチャーの作成
フォームの管理
フォームの作成
アクティブオブジェクトを使用する
アクティブオブジェクトのプロパティ
リストボックス
サブフォームとウィジェット
一覧フォームとレポート
ユーザーフォーム
メソッドの編集
ユーザー＆グループ
メニューとメニューバー
ピクチャーライブラリ
ヘルプTips
リスト
スタイルシート
フィルターとフォーマット
リソースエクスプローラー
レコードの管理
レコードの検索
レコードの並び替え
フォーミュラエディター
クイックレポート
ラベル
データの読み込みと書き出し
Web サービスの公開と使用
データベースのバックアップと復元
Maintenance & Security Center
コンパイル
アプリケーションの仕上げと展開
4Dコンポーネントの開発とインストール
付録

## はじめに


- ようこそ
- 作業環境
- 統合されたサーバー
- 4D v12の新機能
- 4D v13の主な新機能
- 4D v14の主な新機能

## ようこそ

4Dはパワフルなリレーショナルデータベースアプリケーション、ハイパフォーマンスなSQLエンジン、完全にマルチプラットフォームな開発ツール、および動的なWebサーバーが組み込まれています。

4Dを使用して自身のデータを管理したり、異なる種類のデータベース管理タスクを実行するためのカスタムアプリケーションを開発したりできます。
例えば4Dを使用することで以下のようなことが可能です:

- データベーストラクチャーを作成し、テーブルやフィールドを定義する。
- レコードの入力、更新、表示を行うフォームを設計する。
- レコードの検索や並び替えを行う。
- データベース内の情報を使用してレポートやラベルを作成する。
- 4Dデータベースと他のアプリケーション間でデータの読み込みや書き出しを行う。
- データベースをWebに公開する。

4Dの以下の機能を使用すれば従来型のデータ管理タスクを拡張することができます:

- 洗練されたフォームやレポートを数回のクリックだけで作成できるパワフルなフォームウィザード。
- 秘密データを保護するためのパスワードアクセスシステム。
- カスタマイズされたメニュー、ダイアログボックス、ツールバー、ボタンなどを組み込んだカスタムアプリケーションを作成する能力。このアプリケーションをコンパイルして他のユーザーに提供することが可能です。
- データベースから直接Webに公開されるWebサービス機能。
- 完全なプログラミング言語のセット。
- テーブルやフィールドの作成、およびデータベースデータのクエリを行うことのできる、標準に準拠したSQLエンジン。
- 集中化されたアプリケーションの検証および管理ウィンドウ。

初心者の方にもすぐにデータベースを作成してデータの管理を始めていただけます。経験を積んだユーザーは4Dの開発ツールを使用してデータベースをカスタマイズすることができます。さらに経験を積んだ開発者は4Dのパワフルなプログラム言語やSQLを使用して、ファイルの転送や通信、そしてWebなどによる、洗練された機能や能力をデータベースに加えることができるようになります。

# 作業環境

4Dは2つのモードで動作します。一つは**デザインモード**、もう一つは**アプリケーションモード**です。

**デザインモード**

デザインモードではアプリケーションの設計、開発、およびテストを行います。アプリケーションデザインのすべてのアスペクトはデザインモード内で実装されます。デザインモードではテーブルやフィールド作成、テーブル間のリレーション定義、データ入力や表示・印刷のためのフォームを作成、パスワードアクセスシステムの実装、カスタムメニューの作成、データベースオブジェクトへのメソッドの割り当て等を行います。

例えば会社の従業員の情報を追跡したいとします。デザインモードで[Employees]テーブルを作成し、従業員データ (従業員名、肩書き、入社日、給与など) を格納するフィールドをテーブルに追加します。また会社内の部署に関する情報を格納する[Departments]テーブルを作成することもできます。そしてこれらのテーブル間にリレーションを作成し、ある部署で誰が働いているか、またある従業員はどの部署で働いているかを効率的に決定することができるようになります。

デザインモードではさまざまなウィンドウやエディターを使用して、データベースのデータに対して標準の処理を実行できます。データを入力、特定のレコードの検索、読み込みや書き出し、レポートの印刷、ラベルの生成などを行えます。つまりこのモードではデータの入力やレコードの読み込み、メソッドを実行するなどして、データベースの機能をテストすることができます。ここで提供されるエディターはアプリケーションモードでユーザに提供されるのと同じものです。

ユーザーモードでは以下のことを行えます:

- データを格納するテーブルやフィールドを作成。
- テーブル間のリレーションを定義。
- データを入力、表示、印刷、公開するためのフォームを作成。
- データ入力を簡易にしかつ制御するための選択リストを作成。
- データベース処理を自動化するためのメソッドの作成と実行。
- カスタムメニューを作成してメソッドや自動標準アクションを割り当て。
- 複数のプロセスを作成・管理し、同時に複数のデータベース処理を実行。
- 情報へのアクセス制御を行うためのパスワードシステムの設定。
- WebサーバーやSQLサーバーを起動し、外部アプリケーションからデータにアクセス。
- 実行速度の高速化するためにアプリケーションをコンパイル。
- データベースと4Dエンジンを統合してスタンドアロンアプリケーションをビルド。
- データ複製を含む自動バックアップシステムの設定。
- データの入力や更新。
- データの表示や印刷。
- レコードの検索や並び替え。
- レポートやラベル、グラフの作成。
- データの読み込みや書き出し。
- データベースにインストールされた4Dプラグインの利用。

**アプリケーションモード**

アプリケーションモードはカスタムアプリケーションの実行環境です。カスタムアプリケーションは4D上で動作しますが独自のメニューシステムとスクリーンデザインを持ちます。デザインモードで作成されたアプリケーションは通常このモードで配布されます。デザインモードからアプリケーションモードに移動するには、**実行**メニューから**アプリケーションテスト**を選択します。

アプリケーションモード内では、アプリケーションが使用するメニューやフォームからデータの入力・処理・表示に使用されるメソッドまで、開発者がアプリケーションのすべてをコントロールします。メニュー項目を作成してメソッドを割り当てたり、データ入力や更新・検索・並び替え・レポートなどの基本タスクを行う標準アクションを割り当てたりするのは、開発者の責任で行います。4Dが提供する標準のエディター (並び替えやラベル) を使用したり、スクリーンやエディターを自作することもできます。

アプリケーションモードは作成するアプリケーションごとに完全に異なるものになることができます。ユーザー視点からすると、アプリケーションモードは特定の情報管理を行うための完全なアプリケーションです。

4Dが生成するデフォルトのメニューバーには**モード**メニューが含まれていて、そこからデザインモードに戻ることができます。

複数のモードのウィンドウが同時に開かれている場合、それぞれのウィンドウをクリックすることでモードを切り替えることができます。ウィンドウをクリックすると、4Dはこのウィンドウを前面にし、そのウィンドウが属するモードをアクティブにします。

## 統合されたサーバー

4Dには3つのサーバー、データおよびアプリケーションサーバー (4D Server)、Webサーバー、SQLサーバー が統合されています。

これらのサーバーの管理はとても簡単で、最小限の設定しか必要としません。サーバーの実行を制御するための管理コマンドが提供されています。これらのサーバーを使用するためには、適切なライセンスを入手する必要があります。

これらのサーバーの起動と管理については以下のマニュアルで説明されています:

- データおよびアプリケーションサーバー: 4D Server リファレンスマニュアル, OBJECT SET ENTERABLE
- 4D Webサーバー: 4Dランゲージリファレンスマニュアル, Webサーバ設定と接続管理
- 4D SQLサーバー: 4D - SQLリファレンスマニュアル, 4D SQLサーバの設定

## 4D v12の新機能

広く使用されているテクノロジーに対してオープンであり、開発者の生産性と創造性を向上させるための新機能により、4D v12は4D製品ラインの進化の中で新たなステップを刻むものとなりました。

- アーキテクチャーレベルや開発者の作業環境レベルで、デベロッパーコミュニティからのニーズが高かった数多くの機能 (新しいコードエディター、検索と置換機能の向上、PHPスクリプトの利用、UUIDの管理、ピクチャーメタデータの管理、オブジェクトプロパティを管理するコマンドの拡張、そしてXMLやSVGサポートの拡張など) が統合されました。
- 4Dアプリケーションにコンポーネントをインストールする方法がサポートされることで、すべてのデータベースで同じコンポーネントを利用できるようになりました。さらにSQLエクスターナルデータベースを使用することで、コンポーネントからテーブルやフィールドを利用することができるようになりました。
- 新しいPRINT OBJECTコマンドおよびリストボックス印刷のサポート、WindowsでのPDF作成のサポートなど、印刷機能が拡張されました。
- 新しいフォームオブジェクトとウィジェット (ステッパー、リッチテキスト、日付および時間セレクター (日付ピッカー)、検索エリア、そして階層リストボックス) を使用すればモダンで洗練されたインターフェースを作成できます。サブフォームの動作が大きく改良され、特にコンポーネントとして利用することが容易になりました。新しい定義済みオブジェクトライブラリを使用すればこの機能に素早くアクセスできます。
- 最後に、4D SQLランゲージにパワフルな新機能が追加され、データの複製や同期、そしてセッション中で異なる4Dデータベースを開いたり閉じたりすることができるようになりました。

## 4D v13の主な新機能

4D v13は既存のデータベースとの互換を最大限保ちつつ、オープンであることにフォーカスしています:

- 完全に書き直された HTTP サーバー、および 64-bit ODBC ドライバーを使用すれば他のデータベースやインターネットからデータベースにアクセスする事ができます。
- データベース内のすべてのメソッドに読み込み / 書き込みアクセスを可能とする新しいソースツールキットを使用すれば、他の開発ツールを使用してバージョン管理などを行うことができます。ソースツールキットを使用して、例えばコードを標準のテキスト形式で書き出し、管理することができます。
- Web エリアでの描画エンジンの選択、リストボックスの新しいオプション、ピクチャーメタデータのインデックス化、Windows では Direct2D 描画エンジンや XPS を使用したプリントプレビューを使用するなど、新世代のインターフェースを構築できます。
- 最後に、4D v13 は 4D アプリケーションの最適化や開発を容易にする数多くの機能を提供します。BLOB やピクチャーデータを外部ファイルとして保存したり、プログラムからフォームオブジェクトにアクセスする為に使用できる一連のコマンドが追加されました。

## 4D v14の主な新機能

4D v14では、システムアーキテクチャーが大きく進化したことに加え、あなたのアプリケーションのインターフェースをより充実させるための数々の新機能を備えながらも、既存のデータベースとの互換性も最大限確保しています。

- 4D コアの現代化：以前の *carbon* ライブラリーを内部的に書き直すことにより、4D のパフォーマンスはそれらのライブラリーによって課せられていた制約から解き放たれました。その一方で、QuickTime、4D Chart、Pict ピクチャフォーマットといった旧世代のテクノロジーはサポートされなくなっています。
- 4D v14 のフォームには list object (リストボックス、階層リスト、コンボボックス等)、Web areas (JavaScript との親和性の向上、デバッガ等)、そしてテキストエリア(静的テキストの回転、スタイル付テキストの機能拡張) といった分野で新機能がたくさん盛り込まれています。
- 開発環境も、アップデートの手順の簡略化、新しい query ダイアログボックス、デバッグヘルプの拡張、そして新しい On Host Database Event データベースメソッド データベースメソッドなどにより強化されています。
- Wakanda アプリケーションが直接 4D データベースのデータを使用できるように、特定の機能が実装されています。
- 4D のプログラミング言語は様々な新コマンドによってさらに充実したものとなっています。特にオブジェクトプロパティをより自由に扱えるようになったコマンドの進化は顕著で、オブジェクト (C_OBJECT コマンドを参照して下さい)や "JSON" といった新しい型のデータも扱えるようになりました(新テーマも設定されています)。

# 4Dファイルの管理


- 4Dを開始する
- ローカル/リモートアーキテクチャー
- 新しいアプリケーションを作成する
- ローカルデータベースを開く
- リモートアプリケーションに接続する
- プラグインやコンポーネントのインストール
- データファイルを変更する
- 4Dファイルの説明
- 以前のバージョンからのデータベースの変換

## 4Dを開始する

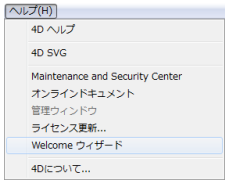
インストールガイドに記載されている通りに4Dをインストールしたら、アプリケーションを開始できます。

これを行うにはアプリケーションアイコンを選択してOSの**ファイル**メニューから**開く**を選択するか、このアイコンを直接ダブルクリックします。

Welcomeウィザードが表示されます:

ウィザードは主に新規の4Dユーザーをガイドするために使用されます。4Dの使用を開始するか、アプリケーションのアクティベーションを行うかなど、利用可能な選択肢が表示されます。

デフォルトでウィザードは毎起動時に表示されます。**起動時にこのダイアログを表示しない**オプションにチェックすると、このダイアログを隠すことができます。この場合、4Dの起動時は環境設定の**一般**ページ**オプション**エリアの"**開始時**"で選択されたアクションが行われます。**ヘルプ**メニューの**Welcome ウィザード**コマンドを選択すると、いつでも表示できます:

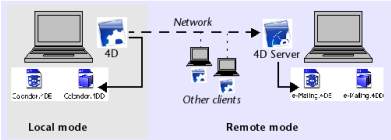



Welcomeウィザードを表示しない場合、4Dは環境設定で設定された起動時オプションの動作を行います (一般ページ参照)。

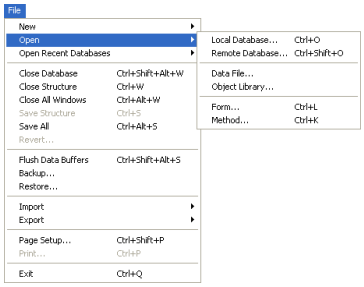
行いたいアクション、または追加の情報を見るために"INFO"ボタンをクリックできます。クローズボックスでウィザードウィンドウを閉じても4Dアプリケーションは起動されたままですが、データベースは開かれていません。この場合、Maintenance and security center (メンテナンスモードでの表示参照) でデータベースを開くか、または**ファイル**メニュー/ツールバーのデータベースを作成/開くコマンドを使用することができます。データベースの作成や開くことに関する情報は新しいアプリケーションを作成するとローカルデータベースを開くを参照してください。

## ローカル/リモートアーキテクチャー

4Dアプリケーションではローカルおよび4D Serverで実行されているリモートのデータベースを開くことができます。前者の場合、アプリケーションはローカルモード (シングルユーザー) で動作します。後者の場合はリモートモード (クライアント/サーバー) で動作します。




開くデータベースを選択する際には、開きたいデータベースのタイプに基づき、使用するモードも選択することになります:




- **開く > ローカルデータベース**: 標準のドキュメントを開くダイアログボックスが表示され、4Dデータベースファイルを選択できます (ローカルデータベースを開く参照)。
- **開く > リモートデータベース**: 公開データベースを選択することのできる、4D Serverに接続ダイアログが表示されます。このダイアログボックスとリモートデータベースを開くことに関する詳細は、4D Server Referenceマニュアルの4D Serverデータベースへの接続を参照してください。

## 新しいアプリケーションを作成する

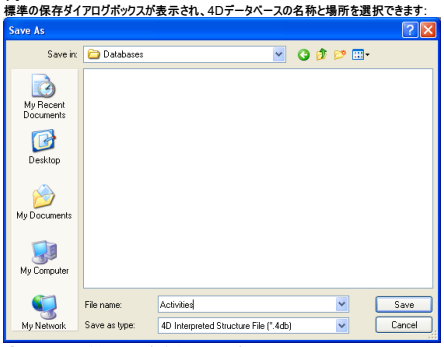
空のデータベースを作成するか、または他のデータベースから書き出したストラクチャー定義からデータベースを作成することができます。

4Dアプリケーションで作成した場合も、4D Serverアプリケーションで作成した場合も、同じ方法でデータベースを作成できます。

### 空のデータベースを作成する

テーブルやフィールドおよびフォームなどを全く含まない空のデータベースを作成できます。

空のデータベースを作成するには以下の通りにします:

1. ファイルメニューまたは4Dツールバーのボタンから**新規 > データベース...**を選択します。ツールバーの"新規"ボタンを直接クリックすることもできます。
   標準の保存ダイアログボックスが表示され、4Dデータベースの名称と場所を選択できます:
   
   
2. データベースの名称を入力し、**保存**をクリックします。
   OSが許可するファイル名を指定できます。
   **注意**: データベースを他のプラットフォームで利用する場合には、そのプラットフォームの規則に従う事を考慮しなければなりません。例えば、バックスラッシュ(¥)記号はOS Xでは問題ありませんが、Windowsでは許されません。

ダイアログボックスを受け入れると、4Dは (データベースが開かれていれば) カレントデータベースを閉じ、指定した名称で指定した場所にフォルダーを作成し、データベースとして動作するために必要なファイル一式をそこに配置します。4Dデータベースのアーキテクチャーに関する詳細はデータベースアーキテクチャーを参照してください。

4Dアプリケーションウィンドウにエクスプローラーが表示されます。ここからプロジェクトフォームを作成したり、ストラクチャーエディターでテーブルやフィールドを追加することができます。

### ストラクチャー定義からデータベースを作成する

XMLフォーマットで書き出されたストラクチャー定義を使用して、同じ構造のデータベースを新規に作成できます。この場合ストラクチャー定義は、必要に応じて複製が可能なストラクチャーテンプレートとみなすことができます。詳細はストラクチャー定義の書き出しと読み込みを参照してください。

XMLストラクチャー定義はそのまま、あるいはXMLエディターで編集して、使用することができます。これはプログラムでストラクチャーを生成するために使用されるいかなるタイプのメカニズムも利用できることを意味します。

さらに4Dストラクチャー定義XMLファイルの内部フォーマットは公開されているので (4Dストラクチャー定義のフォーマット参照)、他のデータベース環境やデザインアプリケーションからこのタイプのファイルを構築し、自動で4Dデータベースを生成することができます。

ストラクチャー定義からデータベースを作成するには以下の手順に従います:

1. **ファイル**メニューまたは4Dツールバーから**新規 > ストラクチャー定義を使用したデータベース...**を選択します。
   標準のドキュメントを開くダイアログボックスが表示され、開く定義ファイルを選択できます。4Dストラクチャ定義の文法に準拠したXMLフォーマットのファイルを選択しなければなりません (プログラムはDTDを使用してファイルを検証します)。
2. ストラクチャー定義ファイルを選択して**OK**をクリックします。
   4Dは作成するデータベースの名称と配置場所を指定するダイアログボックスを表示します。
3. データベースの名称と場所を選択して、**保存**をクリックします。

XMLが妥当であれば、4Dは (開かれていれば) カレントデータベースを閉じ、ストラクチャー定義に基づき新しいストラクチャーを作成します。指定された場所に指定された名称でフォルダーが作成され、データベースとして動作するために必要なファイルがすべて配置されます。そしてエクスプローラーウィンドウが表示されます。

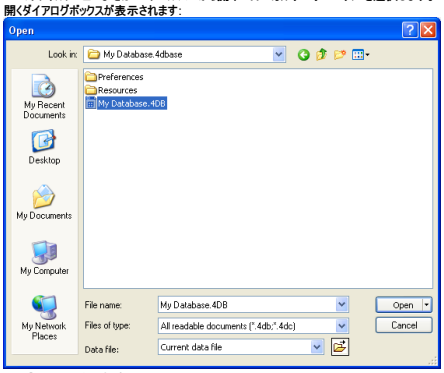
## ローカルデータベースを開く

### 開くダイアログボックス

標準の開くダイアログボックスでは4Dデータベースを選択することができ、いくつかの開始オプションが提供されます。

開くダイアログボックスを使用して既存のデータベースを開くには以下の手順に従います:

1. 4Dのファイルメニューまたは4Dツールバーから**開く > ローカルデータベース...**を選択します。
   開くダイアログボックスが表示されます:



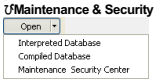

2. 開くデータベースを選択する。
   - Windowsでは、データベースストラクチャーファイルを選択します (拡張子.4DBまたは.4DC)。
   - Mac OSでは、データベースパッケージ (拡張子.4dbase) またはデータベースストラクチャーファイル (拡張子.4DBまたは.4DC) のいずれも選択できます。データベースパッケージに関する詳細は.4dbase拡張子を参照してください。
   - また.4DLinkタイプのアクセスファイルも選択できます。4DLinkファイルにはデータベースを開くために必要なパラメター (アドレスや認証情報など) が格納されています。これはローカルおよびリモートデータベースで機能します。4DLinkファイルの作成と利用については4D Server Referenceマニュアルの4D Serverデータベースへの接続を参照してください。
3. 開始時のオプションを選択し (次の段落参照)、**開く**をクリックしてデータベースを開きます。

**注:** .4dbaseパッケージ、.4DB、.4DCストラクチャーファイル、または.4DLinkファイルをドラッグ＆ドロップやダブルクリックして、開くダイアログを経由せず、4Dアプリケーションを開くこともできます。しかし複数の4Dバージョンがインストールされている環境でファイルをダブルクリックした場合、どの4Dバージョンが開かれるかはOSが決定するため、目的のバージョンでは開かれない危険性があります。

### 開くダイアログオプション

標準のシステムオプションに加え、4Dが提供する開くダイアログボックスには開くとデータファイル、2つのオプションがあります。

- **開く**: このメニューではデータベースを開くモードを指定できます。**インタープリターデータベース**、**コンパイル済みデータベース**、および**Maintenance & Security Center**を選択できます:




  - **インタープリターデータベースとコンパイル済みデータベース**: これら2つのオプションは、選択したデータベースがインタープリターおよびコンパイル済みコードを含んでいる場合に選択可能となります。いずれかのモードが利用できない (データベースがコンパイルされていなかったり、コンパイルコードのみ) 場合、利用可能なオプションのみが選択できます。
  - **Maintenance & Security Center**: 損傷を受けたデータベースに必要な修復を施すために、保護モードでデータベースを開きます。詳細はメンテナンスモードでの表示を参照してください。

  直接**開く**ボタンをクリックすると、データベースは前回開かれたときに使用されたモードで開かれます。

- **データファイル**: このメニューでは、データベースで使用するデータファイルを指定できます。デフォルトで**現在のデータファイル**オプションが選択されています。**開く**をクリックするとデータベースはこのファイルを使用して開きます。ボタンをクリックすると、システムウィンドウでディスク上のこのファイルの位置が表示されます。
  このメニューには他に2つのメニューがあり、4Dの同じストラクチャーを使用して異なるデータファイルを使用することができます。
  - **他のデータファイルを選択**: このオプションを使用すると、カレントではない既存のデータファイルを使用してデータベースを開きます。このオプションを選択して**開く**をクリックすると、標準のデータファイルを開くダイアログが表示され、データファイルを選択することができます。
  - **新しいデータファイルを作成**: このオプションを使用すると、データベースの空のデータファイルを作成できます。このオプションを選択して**開く**をクリックすると、標準のデータファイルを保存ダイアログが表示され、作成するデータファイルの名前と場所を指定できます。

  カレントデータファイルを変更すると、4Dは以降そのデータファイルをデフォルトで開くようになります。データファイルを移動したり名称を変更したりすると、データファイルを再指定する必要があります。

データベースの起動時にキーボードショートカットを使用してデータファイルを変更できます。詳細はデータファイルを変更するを参照してください。
データベースのカレントデータファイルは、いつでもMaintenance and security centerのLIST OF STYLE SHEETSページで確認できます。

### データベースを開くショートカット

4Dは開くダイアログを経由せずにデータベースを直接開くための機能を複数提供しています: ドラッグ＆ドロップ、"最近使用したデータベースを開く"メニュー、そして自動で彩度に使用したデータベースを開く機能です。

- ドラッグ＆ドロップでデータベースを開くには:

  1. 開くデータベースファイルを選択し、4Dアプリケーションアイコンにドロップします。
     - Windowsではデータベースストラクチャーファイル (.4DBおよび.4DC拡張子) を選択します。
     - Mac OSではデータベースパッケージ (.4dbase拡張子) あるいはデータベースストラクチャーファイル (.4DBおよび.4DC拡張子) いずれも選択できます。パッケージアーキテクチャーに関する詳細は.4dbase拡張子を参照してください。
     - 両プラットフォームとも.4DLinkアクセスファイルを選択できます。

- 最近使用したデータベースを開くには:

  1. 4Dの**ファイル**メニューまたは4Dツールバーの開くボタンに割り当てられたメニューから**最近使用したデータベースを開く > データベース名**を選択します。

  最初にアプリケーションが起動されるとき、このメニューは空です。ローカルあるいはリモートのデータベースが使用されるたびに、それがこのメニューに追加されていきます。4Dは使用したデータベースの名称と位置を自動で記録していきます。
  **ファイル**メニューから使用できる**メニュークリア**コマンドを使用すると、メニューからすべてのデータベース名を取り除くことができます。

- 最後に使用したデータベースを自動で開くには:

  1. 環境設定の"一般"ページを開きます。環境設定の表示については環境設定のアクセスを参照してください。
  2. "オプション"エリア内の"開始時"メニューから**最後に使用したデータベースを開く**を選択します。

  アプリケーションが起動されるたびに、4Dは自動で最後に使用したデータベースを開きます。

  **注:** このオプションが選択されているときに開くダイアログボックスを強制的に表示させるためには、アプリケーション起動時に**Alt** (Windows) または **Option** (Mac OS) キーを押します。

## リモートアプリケーションに接続する

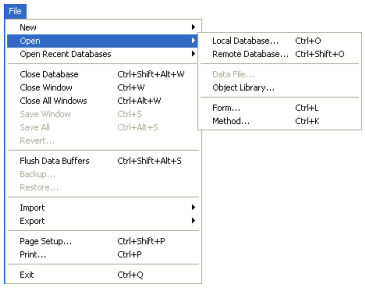
リモート4Dから4D Serverデータベースに接続する方法は3つあります:

- **接続ダイアログボックスを使用する**
- **最近使用したデータベースを開く**メニューを使用する
- データベースへのアクセスパラメタを含む4DLinkショートカットファイルを使用する

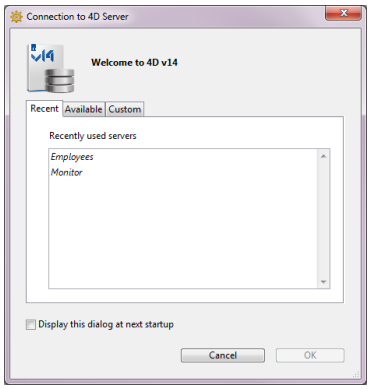
**接続ダイアログボックスを使用する**

4D Server接続ダイアログボックスを表示するには、まず4Dアプリケーションを起動します。

**ファイル**メニューの**開く**コマンド (または4Dツールバーの対応するボタン) を使用して、4Dデータベースを開くモードを選択できます:



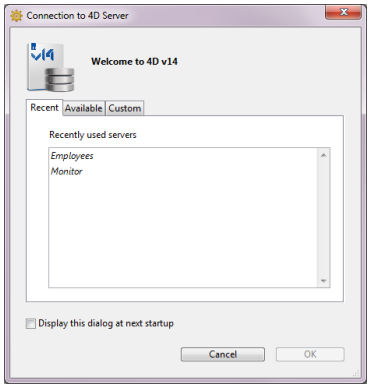

**開く>リモートデータベース...** コマンドを選択
4D Server接続ダイアログボックスが表示されます。このダイアログボックスには最近使用したサーバ、利用可、そしてカスタムの3つのページがあります:




**このダイアログを次回起動時に表示**オプションが選択されていると、4Dアプリケーション起動時に自動でこのダイアログボックスが表示されます。

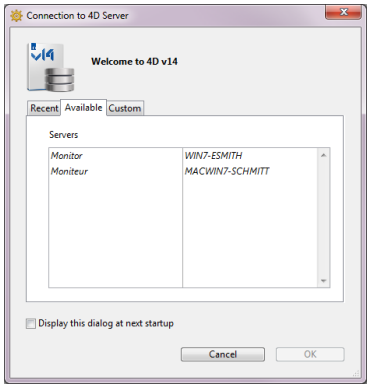
**Note**: 4DのWelcomeダイアログボックスで**4D Serverに接続**をクリックしてもこのダイアログを表示できます。

**"最近使用したサーバ" ページ**




**最近使用したサーバ**ページは、最近使用した4D Serverデータベースを記憶しています。リストは名前順でソートされます。このリストからサーバに接続するには、名前上でダブルクリックするか、名前を選択して**OK** ボタンをクリックします。

**"利用可" ページ**



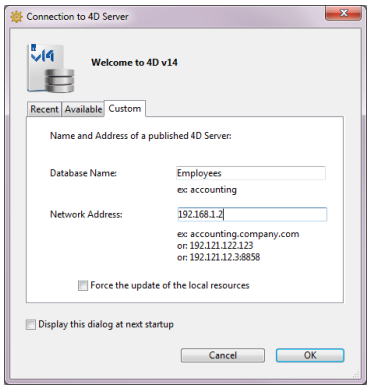

4D Serverには組み込みのTCP/IPブロードキャストシステムがあり、デフォルトで、ネットワーク上に利用可能な4D Serverデータベースの名前を公開します。この名前は接続ダイアログボックスの**利用可**ページに表示されます。

このリストは、名前が見つかった順に表示され、動的に更新されます。このリストからサーバに接続するには、名前上でダブルクリックするか、名前を選択して**OK** ボタンをクリックします。

**Notes:**

- 暗号化オプションが有効で公開されているデータベース名の前にはキャレット (^) が置かれます。詳細はクライアント/サーバ接続の暗号化を参照してください。
- ネットワーク上へのデータベース名の公開を禁止できます (公開 (環境設定)参照)。この場合、このページからの接続はできません。カスタムページを使用します。

#### "カスタム" ページ




**カスタム**ページではIPアドレスでネットワーク上のサーバを指定し、それに任意の名前をつけられます。

4D ServerのTCP/IPブロードキャストシステムをカスタマイズして、ネットワーク上に自動ではサーバデータベースの名前を公開しないようにできます (公開 (環境設定)参照)。この場合、サーバデータベース名は"利用可"ページに表示されません。しかし、ブロードキャストされていないサーバデータベースのIPアドレスを知っていれば、そのIPアドレスを入力できます。

- **データベース名**: 4D Serverデータベース名を指定できます。この名前は**最近使用したサーバ**ページでデータベースを参照する際に使用されます。
- **ネットワークアドレス**: 4D Serverが起動されたマシンのIPアドレスを指定します。2つのサーバが同時に同じマシン上で起動されているときは、IPアドレスの後にコロンとポート番号を続けます。例えば: 192.168.92.104:19820
  デフォルトで、4D Serverの公開ポートは19813です。この番号はアプリケーションの環境設定で変更できます (設定 (環境設定))。

**Note: カスタム**タブをクリックするときに**最近使用したサーバ**や**利用可**ページで項目が選択されていた場合、これら2つのフィールドには対応する情報が表示されます。

このページでサーバを指定したら、**OK**ボタンをクリックしてサーバに接続できます。接続したサーバは**最近使用したサーバ**のリストに加えられます。

**Note:** データベースが暗号化されて公開されている場合、名前の前にキャレット (^) を置かなければなりません。そうでなければ接続は拒否されます。詳細はクライアント/サーバ接続の暗号化クライアント/サーバ接続の暗号化を参照してください。

#### ローカルリソースを強制的に更新

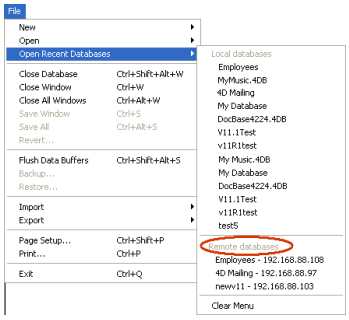
このオプションを選択すると、接続時にクライアントマシン上のローカルリソースを機械的に更新します。ローカルリソースはそれぞれのクライアントマシンに格納される、データベースに関連したストラクチャ情報です。

ルール上ローカルリソースの更新は、リモートマシンが接続したときに、データベースのストラクチャが前回から更新されていれば、自動で行われます。ほとんどの場合、このオプションは必要ありません。特別なケースで更新を強制したい時にこのオプションを使用します。

### 最近使用したデータベースを開くメニューを使用する

**最近使用したデータベースを開く**メニューコマンドは、以前に接続したことのある4D Serverデータベースに直接接続するために使用します。

このコマンドは4Dの**ファイル**メニューにあります。4Dを使用してローカルデータベースおよびリモートデータベースを開いている場合、このメニューには両タイプのデータベースがリストされます。リモートデータベースはメニューの下部にリストされます:




データベース名の隣にはIPアドレスが表示されます。

**メニュークリア** コマンドでメニューをリセットできます。

### 4DLinkファイルを使用する

4Dデータベースを自動で簡単に開くために使用するアクセスファイルを生成できます。通常アクセスファイルは4D Serverリモートデータベースの IPアドレスと接続識別子を保存し、ユーザが行うべき作業を取り除きます。

アクセスファイルを使用してローカルデータベースを開くこともできます。

#### ファイルの作成

4Dデータベースの接続ファイルはXML ファイルで、".4DLink" 拡張子が付けられます。4Dは"最近使用したデータベース" のサブメニューを構築するために、このタイプのファイルを生成して利用します。ローカルデータベースを初めて開くとき、またはサーバに接続するとき、4D は.4DLink ファイルを自動で生成します。

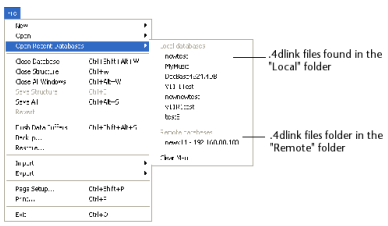
4Dが自動で作成した.4DLink ファイルは、ユーザのローカルな環境設定フォルダに置かれます。このフォルダには、2 つのディレクトリLocal とRemote が作成されます。Local フォルダにはローカルデータベースへの接続に使用できる".4DLink" ファイルが、Remote フォルダにはリモートデータベースへの接続に使用できる"4DLink" ファイルが置かれます。

ローカル環境設定フォルダは以下の場所にあります:

- Windows 7 以降: C:\Users*UserName*\AppData\Roaming\4D\Favorites v*XX*\
- OS X: Users/*UserName*/Library/Application Support/4D/Favorites v*XX*/

*XX* はアプリケーションのバージョン番号を意味します(例えば、4D v14なら"Favorites v14"となります)。

このディレクトリに置かれたファイルは、4D の**ファイル**メニューの**最近使用したデータベースを開く**> サブメニューに表示されます:




".4DLink" ファイルはXML エディタを使用して作成し、接続ID (ユーザ名やパスワード) やデータベースを開くモードなどをカスタマイズした情報を含めることもできます。

".4DLink" ファイルを構築するために使用できるXML キーを定義したDTD が4D より提供されます。このDTD は**database_link.dtd**という名前で、4D の**/Resources/DTD**サブフォルダに見つけることができます。

#### ファイルを使用する

.4DLinkアクセスファイルを使用して4Dアプリケーションを起動し、目的の4Dデータベースを開くことができます。ファイルを使用する方法は2つあります:

- ダブルクリック、または4Dアプリケーションにドラッグ&ドロップ
- **最近使用したデータベースを開く**サブメニュー (ローカル環境設定フォルダ内のファイル)
  "リモート"データベースタイプの.4DLinkファイルを他のマシンにコピーして使用できます。

**Note**: 4Dと4D Serverの接続ダイアログで4DLinkファイルを選択することも可能です (ローカルデータベースの開始のみ)。

## プラグインやコンポーネントのインストール

プラグインやコンポーネントを使用するとアプリケーションに機能を追加することができます。

- プラグインは外部プログラムであり、通常はC言語を使用して開発され、さまざまな機能を提供することができます。例えば4D Writeプラグインはワープロソフトウェアの機能を追加します。4D Internet Commandsなど特定のプラグインは無料で提供されます。あるいは追加のライセンス料が必要なものもあります。プラグインは4D SASあるいはサードパーティーから提供されます。また開発者が自分で作成することも可能です。
- コンポーネントは4Dで開発された独立したプログラムで、ハイレベルな機能を提供するものです。例えば4D SVGコンポーネントはSVGを利用する4D標準の機能を拡張します。コンポーネントも開発者が自分で作成できます。

プラグインとコンポーネントは4Dおよび4D Server環境の適切な場所にインストールして使用します。利用形態に応じて異なる場所にインストールできます。

**注:** コンポーネントの開発と配布については4Dコンポーネントの開発とインストールを参照してください。

**インストールの手順**

プラグインやコンポーネントは4D環境の適切なフォルダーにコピーすることでインストールされます。プラグインは**PlugIns**フォルダーに、コンポーネントは**Components**フォルダーに配置します。これらのフォルダーは必要に応じて異なる2つの場所に配置できます (次節参照)。

フォルダーに配置するものは以下の通りです:

- “PluginName.bundle” フォルダー (Mac OSではパッケージと呼ばれます)。このフォルダーにはWindowsおよびMac OS用の4Dプラグインが含まれます。この構造により、4D Server環境において、接続するクライアントのOSに応じたプラグインがロード/配信され、クライアント上で実行されます。プラグインをインストールするには、“PluginName.bundle”フォルダー (パッケージ) をPlugInsフォルダーに配置します。

- コンポーネントは拡張子が.4db (インタープリターマトリクスデータベース)、.4dc (コンパイル済みマトリクスデータベース) あるいは.4dbase (パッケージタイプマトリクスデータベース、.4dbase拡張子参照) のファイルであるマトリクスデータベースです。マトリクスデータベースのエイリアス (Mac OS) やショートカット (Windows) を使用できます。Componentsフォルダーにはコンポーネントの動作に必要なファイル (XLIFFや画像など) やフォルダーを配置できます。他方、プラグインやコンポーネントのサブフォルダー、データファイル、ユーザーストラクチャーファイル (.4DA) などは配置できません。これらのファイルが存在しても4Dは無視します。コンポーネントアーキテクチャーに関する詳細は概要を参照してください。

プラグインやコンポーネントは4D起動時にロードされるので、これらをインストールする際には4Dアプリケーションを終了する必要があります。
インストールが終了したら4Dでデータベースを開きます。プラグインの利用に特別なライセンスが必要な場合、ライセンスをインストールするまで、プラグインはロードされますが、使用することはできません。

**PlugInsおよびComponentsフォルダーの場所**

PlugInsやComponentsフォルダーは2つの異なる場所に配置できます:

- **4D実行アプリケーションレベル**:
  - Windows: .exeファイルと同階層
  - Mac OS: アプリケーションパッケージ内のContentsフォルダーの直下。

  この場合、プラグインやコンポーネントはこのアプリケーションで開かれるすべてのデータベースから利用できます。

- **データベースストラクチャーファイルと同階層**:
  この場合、プラグインやコンポーネントはそのデータベースでのみ利用可能です。

場所の選択はプラグインやコンポーネントをどのように使用するかによって決定します。

同じプラグインやコンポーネントが両方の場所にインストールされている場合、4Dはストラクチャーファイルと同階層にインストールされたもののみをロードします。コンパイルされ4D Volume Desktopがマージされたアプリケーションでは、同じプラグインやコンポーネントのインスタンスが複数存在する場合、アプリケーションを開くことができません。

**Windows環境下における 64-bit プラグインに対するサポート**

v14では、64-ビットWindows版のプラグインも用意されています。これを使用することによって、In v14, 4DServerの64-ビットWindows版の環境設定の中で64-ビットアーキテクチャの恩恵にあずかれることができます。

64-ビット版はプラグインバンドル の中に含まれています。特別なインストール操作は必要ありません。

64-ビットWindows版では以下のプラグインがご利用いただけます:

- 4D Write
- 4D View
- 4D Internet Commands
- 4D Pack
- 4D ODBC Pro
- 4D For OCI (64-bit 4D For OCI プラグインをインストールする前に、64-bit OCI をインストールする必要があります。このバージョンのOCIはOracle, Inc.からダウンロードすることができます。)

32-ビット版のプラグインと 64-ビット版のプラグインの機能は基本的には全く同じです。ただし、Open external window コマンドは64-ビット版 4D Serverでは使用できないことに注意して下さい。

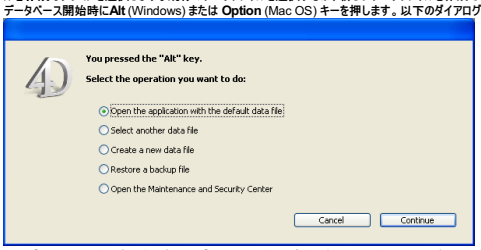
## データファイルを変更する

必要であれば、データファイルを変更することができます。つまり一つのストラクチャーで異なるデータファイルを使用することができます。しかし同時に開くことのできるデータファイルは一つだけです。

データファイルとストラクチャーファイルは対でなければなりません。データ整合性を保持するために、カレントストラクチャファイルで作成されていないデータファイルを開くことはできません。4Dはデータファイルとストラクチャーファイルの作成時またはデータベースの変換時に自動で内部リンク番号 (UUID) を割り当てます。これらの番号はデータファイルが開かれるときに検証されます。

データファイルを変更するには以下の方法があります:

- データベースを開く際に、データベースを開くダイアログボックスの"データファイル"メニューから**他のデータファイルを選択**または**新しいデータファイルを作成**オプションを選択します。既存のデータファイルを選択するか、新しいデータファイルを作成できます (ローカルデータベースを開く参照)。
- データベース開始時に**Alt** (Windows) または **Option** (Mac OS) キーを押します。以下のダイアログボックスが表示されます:




**他のデータファイルを選択**または**新しいデータファイルを作成**を選択して**OK**をクリックします。
- データベースを開いた後、**ファイル**メニューまたは4Dツールバーの"新規"や"開く"ボタンに割り当てられたメニューから、**新規 > データファイル...** または **開く > データファイル...** を選択します。

いずれのケースでもダイアログボックスで既存のデータファイルを選択したり、新しいデータファイルを作成したりできます。新しいデータファイルの作成を選択すると、4Dは元のストラクチャーでデータベースを開きますが、レコードはありません。

他のデータファイルを使用するか新しいものを作成すると、そのデータファイルが新しいカレントデータファイルとなり、次回データベースを開く際も自動で使用されます。
データファイルの名称を変更したり移動したりした場合、手作業でそれを指定する必要があります。

## 4Dファイルの説明

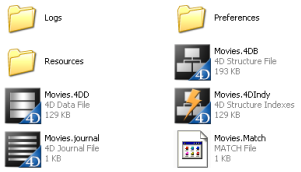
4Dはアプリケーションごとにデフォルトで複数のファイルやフォルダーを作成します。以前のバージョンから変換されたアプリケーションではさらに追加のファイルが使用されます。

**データベースアーキテクチャー**

データベースを作成すると、以下のファイルやフォルダーがディスク上にデフォルトで作成されます:

- **ストラクチャーファイル** (".4DB"拡張子)
  ストラクチャーファイルにはデータベースストラクチャー (テーブル、フィールド、フィールドプロパティ)、フォーム、メソッド、メニュー、パスワード、および選択リストに関連するすべての定義が含まれます。ストラクチャーファイルの名前は、データベースを作成する際にダイアログに入力した名称に".4db"拡張子を付けたものになります。
- **データファイル** (".4DD"拡張子)
  データファイルにはレコードとして入力されたデータと、レコードに属するデータが格納されます。データファイルの名前は、データベースを作成する際にダイアログに入力した名称に".4dd"拡張子を付けたものになります。
  4Dストラクチャーファイルを開く際、アプリケーションはデフォルトでカレントデータファイルを開きます。このファイルの場所や名称を変更した場合はデータファイルを開くダイアログボックスが表示され、使用するデータファイルを選択するか、新しいデータファイルを作成できます (データファイルを変更する参照)。
  不正な処理を避けるため、データファイルは自動でそれを作成したストラクチャーファイルと結び付けられます。
- **ストラクチャーインデックスファイル** (".4DIndy"拡張子) と**データインデックスファイル** (".4DIndx"拡張子)
  4Dデータベースで作成されるインデックスは別ファイルとして作成されます。これらのファイルは自動でストラクチャーファイルと同階層におかれます。これらを移動したり名称変更したりすることはできません。これれを行うと、4Dは自動でインデックスファイルを再作成します。
  *DatabaseName.4DIndx*にはデータインデックスが、*DatabaseName.4DIndy*にはストラクチャーインデックス (特にストラクチャー内を検索する際に使用されます) が含まれます。この構造の利点として、インデックスが壊れた場合に、4Dを起動する前にファイルを物理的に削除しておけば、新しいインデックスファイルが自動で新しく作成されることが挙げられます。
- **データログファイル** (".journal"拡張子) — データベースがデータログファイルを使用する場合のみ作成されます。
  ログファイルは2つのバックアップ間でのデータベースデータの保護を確実なものにするために使用されます。データベースのデータに対して実行されたすべての処理が、このファイルに順番に記録されます。つまりデータ上に処理が行われるたびに、2つの処理、データベースデータ上の処理とログファイル上の処理が同時に行われます。ログファイルはユーザーの処理を妨げたり遅くしたりすることなく、独立して構築されます。データベースは同時に1つのログファイルしか使用できません。
  ログファイルにはレコードの追加/更新/削除やトランザクションなどの処理が記録されます。このファイルはデータベースが作成される際にデフォルトで作成されます。詳細はログファイルの管理を参照してください。
- **Resourcesフォルダー**: 外部リソースを格納するフォルダーです。
  Resourcesフォルダーはデータベースストラクチャーファイル (.4dbまたは.4dc) と同階層になければならず、データベースインターフェースに必要となるファイルを格納するためのものです。
  このフォルダーにはアプリケーションインターフェースの翻訳やカスタマイズに必要なファイル (画像、テキスト、XLIFFファイルなど) を配置できます。
  4Dは特にXLIFFファイルや画像ファイルにおいて、自動のメカニズムを使用してこのフォルダーの内容を扱います。
  リモートモードにおいては、**Resources**フォルダーを使用してサーバーマシンとすべてのクライアントマシン間でファイルを共有できます。詳細は4D Server Referenceマニュアルのリソースフォルダの管理を参照してください。
- **Preferencesフォルダー**: データベース固有の設定ファイルを格納するフォルダーで、必要な場合にのみ作成されます。
  Preferencesフォルダーにはデータベースの設定ファイルが通常XMLフォーマットで格納されます。フォルダーは特にバックアップ機能を使用する際 (*Backup.xml*ファイルを格納するために) またはアプリケーションビルド機能を使用する際 (*BuildApp.xml*ファイルを格納するために) 、自動で作成されます。
- **Logs** フォルダーにはカレントデータベースのログが格納されます。このフォルダーはストラクチャファイルと同階層に作成されます。以下のログファイルが格納されます:
  - データベース変換
  - Webサーバーリクエスト
  - データ検証と修復
  - ストラクチャー検証と修復
  - バックアップ/復元アクションのジャーナル
  - コマンドデバッグ
  - 4D Serverリクエスト (クライアントマシンおよびサーバー上で作成)

  これらのファイルやフォルダーはデータベースの作成時に同じフォルダー内に配置されます。4Dの自動的なメカニズムに影響を与えないために、これらのファイルやフォルダーの名称を変更しないでください。




**変換されたデータベースにおける追加のファイル**

v11より前の4Dで作成されたデータベースを変換すると、追加のファイルが作成されます:

- **ストラクチャーリソースファイル** (".RSR"拡張子)
  このファイルにはデータベースストラクチャーで利用される、古いMacintoshタイプのリソースが格納されています。
  .4DBと.RSRファイルは同じディレクトリ内に同じ名前で存在しなければなりません。そうでなければデータベースを開くことができません。
- **データリソースファイル** (".4DR"拡張子)
  このファイルにはデータベースのデータで利用される、古いMacintoshタイプのリソースが格納されています。

データベースの変換に関する詳細は、DELETE USERを参照してください。

**.4dbase拡張子**

デフォルトで4Dデータベースは自動で**.4dbase**拡張子のフォルダーを作成します。例えば"Invoices"という名称のデータベースは*Invoices.4dbase*というフォルダーの中に作成されます。このフォルダーにはデータベースが動作するために必要なすべてのファイルが格納されます。

このため、Mac OSにおいてデータベースフォルダーはパッケージとして表示されます。パッケージやストラクチャーファイル、あるいはデータファイルをダブルクリックして4Dを起動することができます。ただし複数のバージョンの4Dがインストールされた環境では、この方法でどのバージョンの4Dが起動されるかをOSが決定するため、推奨できません。
パッケージを4Dアプリケーションアイコンにドラッグ＆ドロップして起動することもできます。
Windowsでは.4dbaseフォルダーはフォルダーとして扱われます。

環境設定の"一般"ページにある**新規データベースでパッケージを作成する**オプションを外すと、このデフォルト動作を無効にできます。

## 以前のバージョンからのデータベースの変換

以前のバージョン(11.x, 12.x そして 13.x 等)の4Dまたは4D Serverで作成されたデータベースは4D v14と互換性があります (ストラクチャーおよびデータファイル)。

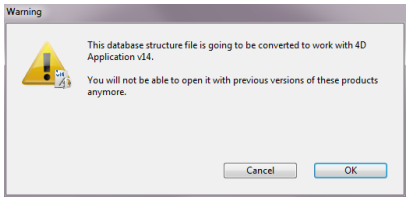
- バージョン11より前のデータベースはウィザードを使用して変換する必要があり、一度変換するとオリジナルのバージョンでは開けなくなります。
- バージョン11や12のデータベースファイルは直接v14用に変換されます。一度変換するとストラクチャーファイルはオリジナルのバージョンでは開けなくなります。他方データファイルについては後述する条件のもと、再びバージョン12で開くことができます。
- バージョン13のストラクチャーファイルは直接v14用に変換され、一度変換するとオリジナルバージョンでは開けなくなります。しかしながら、バージョン13のデータファイルはv14において変換なしで開くことができ、この場合はその後再び13にて開くことが可能です。

**注:** どんなインタープリターストラクチャーも変換することができます。ファイルにコンパイルコードが含まれていてもかまいませんが、変換後に再度コンパイルする必要があります。

v14以前のデータベースでプライマリーキーがないものを変換する場合、警告ダイアログボックスが表示されます。この点については、以下の「プライマリーキーエラーウィンドウ」の章にて詳細な説明があります。

### Databases in version 13

v13で作成されたデータベースは、4D v14で開くときに直接変換されます。その際、警告ダイアログボックスが表示され、ストラクチャーファイルが変換されて旧バージョンでは開けなくなるメッセージが表示されます。




しかしながら、データファイルは変換すことなく開くことができるので、v14で開いたあともv13で再度開くことができます。

### バージョン11/12のデータベース

バージョン11や12のデータベースは、ストラクチャーファイルを4D v14で開く際に直接変換されます。2つのダイアログが連続して表示され、変換されるファイルが以前のバージョンで開けなくなることを警告します。

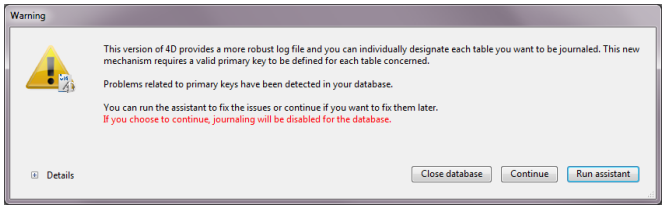
データファイルが変換される際には、インデックスも再構築されます。

**注:** 古いバージョンのデータベースの変換についての詳細は、4D v13 のドキュメント内の Converting databases from previous versions を参照して下さい。

### プライマリーキーエラーウィンドウ

4D v14以降、ログファイルの使用には、ログファイルを使用するためにはログを取るテーブルが全て有効なプライマリーキーを持っている必要があります (ログファイルの管理 を参照して下さい)。

データベース内に、有効なプライマリーキーをないテーブルが少なくとも一つある場合にはエラーダイアログボックスが表示されます。

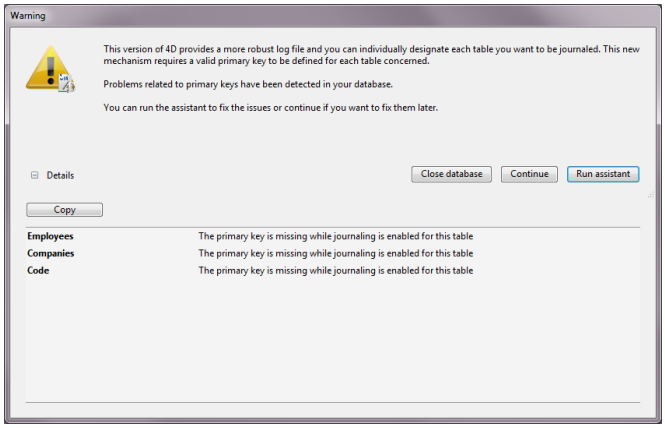



このダイアログボックスは以下のタイミングで現れます:

- v14のデータベースへと変換した後。
- v14のデータベース内で**ログファイルを使用**のオプションにチェックを入れた後。

**注:** データログがデータベース内で有効化されていないとき、エラーダイアログボックスはプライマリーキーエラーに関わらずデータベースを開くというオプションを提供します。

データベーストラクチャーにアクセスできる場合、ダイアログボックスはいくつかの選択肢を表示します。またボックス下部を展開すると、プライマリーキーを持っていないテーブルの一覧が表示されます:




その後選択できるオプションは以下の通りです:

- **コピー**: ウィンドウからの情報を解析のためにクリップボードへコピーします。
- **データベースを閉じる**: データベースには何も触れずに閉じます。
- **続ける**: エラーを処理せずにデータベースを開きます。この場合、データベースのログは無効になり(**ログファイルを使用**のオプションのチェックが外れます)、データベースはv14で正常に使用できます。データベースにてログファイルを使用したくない場合に有用です。
- **アシスタントを起動**: プライマリーキー管理ウィンドウを表示し、データベース内の全テーブルを更新します。データベースを更新する場合にはこのオプションの仕様が推奨されます。アシスタントについては プライマリーキー管理 のセクションで詳細な説明があります。

データベーストラクチャーへのアクセスがない場合、データログを有効化することはできません。データベース管理者へと連絡を取ることを勧めるメッセージが表示されます。

### バージョン11/12のコンポーネント

4D v14はv13やv12やv11のコンポーネントを、変換しなくても開くことができます (コンパイルおよびインタープリター)。コンポーネントは常に読み込みのみモードで開かれる点に留意してください。

コンポーネンツのリコンパイルは不要ですが、v14への変換は.4DBファイルのみ可能です。.4DCファイルはアップグレードできません。

# 環境設定


概要

一般ページ

ストラクチャーページ

フォームページ

メソッドページ

ショートカットページ

## 概要

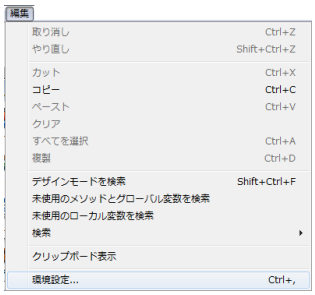
環境設定ではあなたの作業環境のデフォルト動作オプションを指定できます。例えばメソッドエディターで使用されるカラー、自動フォーム作成オプション、フォームエディタにデフォルトで表示される項目などです。これらの設定はこのマシン上で4Dや4D Serverを使用して開かれるすべてのデータベースに適用されます。

**4D Server:** 同時に複数のユーザーが環境設定ダイアログボックスで設定を更新しようとすると、オブジェクトのロックが発生します。一度に一人のユーザーのみが環境設定ダイアログボックスを使用できます。

**注**: 4Dはデータベース固有の設定を行うデータベース設定ダイアログも提供しています。詳細はデータベース設定を参照してください。

### アクセス

環境設定ダイアログボックスにアクセスするには**編集** (Windows) または**4D**アプリケーションメニュー (Mac OS) から**環境設定...**を選択します:




このメニューコマンドはデータベースが開かれていない場合でも利用できます。

OPEN SETTINGS WINDOWコマンドや"環境設定"標準アクションを (ボタンやメニューに割り当てて) 使用すればアプリケーションモードでも環境設定ダイアログボックスを表示できます。

### 設定の保存

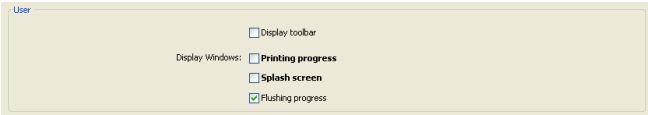
環境設定ダイアログに保存された設定はXMLフォーマットで**4D vXX Preferences.4DPreferences**という名称のファイルに保存されます。このファイルはカレントユーザーの環境設定フォルダーに保存されます:

- Windows XP: {disk}\Documents and Settings\{UserName}\Application Data\4D
- Windows Vista/Windows 7: {disk}\Users\{UserName}\AppData\Roaming\4D
- Mac OS: {disk}:Users:{UserName}:Library:Preferences:4D

**注:** *XX*は4Dのバージョン番号です。例えばバージョン12の場合、ファイル名は**4D v12 Preferences.4DPreferences**となります。

### パラメーターのカスタマイズと初期設定

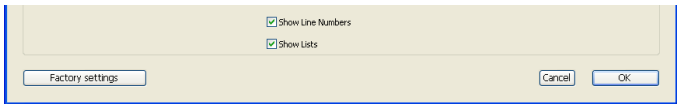
環境設定および設定ダイアログボックスでは、変更された設定内容は**太字**で表示されます:




環境設定内ではダイアログボックスで直接カスタマイズされるか、変換されたデータベースの場合以前のバージョンで変更された設定が、変更箇所として扱われます。

パラメーターは手作業でデフォルト値に置き換えられたときにも太字で表示されます。このように、カスタマイズされたパラメーターはすべて目視で識別することができます。

パラメーターをデフォルト値に戻し、カスタマイズされたことを示す太字スタイルを取り除くためには、**初期設定**ボタンをクリックします:




このボタンをクリックすると、現在表示されているページのすべてのパラメーターがリセットされます。このボタンは現在のページで最低一つのパラメーターが変更されるとアクティブになります。

## 一般ページ

このページでは4Dアプリケーションの一般的な動作を設定するためのオプションが含まれています。

### オプション

このエリアのオプションを使用して、起動時やアプリケーションモードに移動する際の自動機能を設定できます。

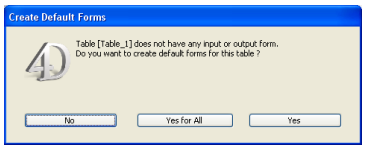
#### 開始時

このオプションではユーザーがアプリケーションのみを起動したときの起動時に4Dが提供するデフォルトの表示を設定します。

- **何もしない**: アプリケーションウィンドウのみが表示されます。
- **ローカルデータベースを開くダイアログ**: 4Dは標準のドキュメントを開くダイアログボックスを表示し、ローカルのデータベースを選択することができます。
- **最後に使用したデータベースを開く**: 4Dは直接最後に使用されたデータベースを開きます。データベースを開くダイアログボックスは表示されません。
  **注**: このオプションが選択されているときに、データベースを開くダイアログボックスを表示させるには、**Alt** (Windows) または **Option** (Mac OS) キーをデータベースを起動する際に押します。
- **リモートデータベースを開くダイアログ**: 4Dは4D Serverに接続する標準のダイアログボックスを表示し、ネットワークに公開されたデータベースを指定することができます。
- **Welcomeウィザードダイアログ** (初期設定): 4DはWelcomeウィザードダイアログボックスを表示します。

#### 自動フォーム作成

このメニューを使用して、ストラクチャーエディターでテーブルを作成した後に例えばレコードを表示するウィンドウを表示する際の、4Dの動作を設定できます。デフォルトで4Dは、新しいテーブルのフォームがまだ作成されていないことを通知し、自動でフォームを作成するかどうかの選択を促すダイアログを表示します:




3つの選択肢があります:

- **しない**: 警告ダイアログボックスは表示されず、デフォルトフォームは作成されません。
- **その都度指定**: テーブルにフォームが作成されていない場合、警告ダイアログボックスが表示されます (初期設定)。
- **常にすべて作成する**: 警告ダイアログボックスは表示されず、自動ですべてのテーブルにデフォルトフォームが作成されます。

#### アプリケーションモードに移動するときに、デザインモードを終了する

このオプションが選択されていると、**アプリケーションテスト**コマンドを使用してユーザーがアプリケーションモードに移動する際に、デザインモードのすべてのウィンドウが閉じられます。このオプションがチェックされていないと (初期設定)、デザインモードのウィンドウはアプリケーションモードの後ろに表示されたままとなります。

### 新規データベース作成時

このグループのオプションは新規にアプリケーションを作成する際に適用されます。

#### 自動バックアップをすべての新規データベースで有効にする

このオプションをチェックして、最低限の保護を行うために、新規データベースのバックアップパラメター (自動のバックアップやログファイル) を設定できます。このオプションについては自動バックアップを参照してください。

#### 新規データベースでパッケージを作成する

このオプションがチェックされていると、4Dデータベースは自動で.4dbase拡張子が付いたフォルダーに作成されます。
この原則のため、Mac OSではデータベースフォルダーが特別なプロパティ付きのパッケージとして表示されます。Windowsではこれは普通のフォルダーと変わりありません。詳細については4Dファイルの説明を参照してください。

#### テキスト比較の言語

このパラメターは新規データベース作成時に、文字列の処理と比較で使用されるデフォルトの言語を設定します。言語の選択は直接テキストの並び替えや検索、大文字小文字の比較ルール等に影響を与えます。ただしテキストの翻訳や日付/時刻/通貨のフォーマットはシステムランゲージ設定が使用され、この設定には影響されません。初期設定では、4Dはシステムに設定されたカレントのユーザー言語を使用します。
4Dデータベースはシステムの言語とは異なる言語で動作することができます。データベースが開かれるとき、4Dエンジンはデータファイルに使用されている言語を検知し、(インタープリターやコンパイルモードの) ランゲージに提供します。データベースエンジンあるいはランゲージいずれでテキスト比較が行われるかに関わらず、同じ言語が使用されます。
新規にデータファイルを作成する際、4Dはこのメニューで設定された言語を使用します。ストラクチャーの言語と名時でない言語のデータファイルを開くと、データファイルの言語が使用され、ストラクチャーに言語コードがコピーされます。

**注:** データベース設定を使用して、開かれているデータベースの言語を変更することができます (テキスト比較参照)。

### ドキュメントの場所

このエリアでは、カレントのブラウザーに表示される4D HTMLドキュメントへのアクセスを設定できます:

- ユーザーがエクスプローラーのOBJECT SET RGB COLORS上のコマンドをダブルクリックしたとき
- ユーザーがメソッドエディター上でコマンド名をクリックし、**F1**キーを押したとき (参照)

4Dオンラインドキュメントサイト (*4D Doc Center)* に直接アクセスするか、ローカルに保存したファイルにアクセスするかを選択できます。

#### ローカルフォルダー

スタティックなHTMLドキュメントの場所を指定します。デフォルトでこれは\Help\Command*language* サブフォルダーに設定されています。このエリアに割り当てられているメニューをクリックすると、場所を見ることができます。このサブフォルダーが存在しない場合、場所は赤で表示されます。
この場所は変更することが可能です。HTMLドキュメントはこのとなるボリュームやCD-ROM上などに置くことも可能です。他の場所を指定するには、メニューの隣の[...]ボタンをクリックし、ドキュメントのルートフォルダー (*fr, en, es, de* または *ja*などの言語に対応するフォルダー) を選択します。

**注:** HTMLドキュメントのダウンロードに関する情報は4D Doc Centerのホームページを参照してください。

#### Webサイト

4D Doc Center上の対応するバージョンのオンラインドキュメントにアクセスするためのURL。4DはこのURLをベースとしてドキュメントの呼び出しを構築します。このURLを変更して、例えばアプリケーションランゲージとは異なる言語のドキュメントを参照できます。エリア右側のテストボタンを使用すると、デフォルトのブラウザーが起動され、指定されたURLにアクセスします。

#### 最初にローカルフォルダーを見る

(デフォルトでチェックされている) このオプションは、エクスプローラーやF1キーで呼び出されたドキュメントのページを探す場所を設定します。

- チェックされている場合、4Dはまずローカルフォルダーでページを探します。ページが見つかれば、4Dはそのページをカレントブラウザーで表示します。見つからない場合、4Dは自動でWebサイトのオンラインドキュメントを参照します。この場合インターネットに接続されていない環境でも、ローカルのドキュメントが参照できます。
- チェックされていない場合、4DはWebサイトのオンラインドキュメントに直接アクセスし、カレントブラウザーでページを表示します。ページが見つからない場合、4Dはブラウザーにエラーメッセージを表示します。このオプションはマシンがインターネットに接続されている場合にのみ使用できます。

## ストラクチャーページ

### プライマリーキー

データベース設定内のこれらのオプションによって、新しくテーブルが追加されたときに4Dまたは プライマリーキー管理 によって自動的に追加されるプライマリーキーのデフォルトの名前と型を変更することができます。

以下のオプションが設定できます:

- **Name** (デフォルトでは"ID"): プライマリーキーのフィールドのデフォルトの名前を設定します。標準的な4Dテーブルの命名規則に従う範囲内であればどんな名前も使用できます (テーブルとフィールドの命名規則 を参照して下さい)。
- **デフォルトタイプ** (デフォルトでは倍長整数): プライマリーキーフィールドのデフォルトのタイプを設定します。UUIDを選択することもできます。この場合、デフォルトで作成されたプライマリーキーフィールドは文字型であり、UUIDフォーマットと 自動UUID プロパティにチェックが入っています。

### ストラクチャーエディター

このオプショングループでは、4Dストラクチャーエディターの表示方法を設定します。

#### ストラクチャーの描画クォリティ

このオプションでストラクチャーエディターの描画レベルを変更できます。デフォルトで品質は**高**に設定されています。**標準**品質を選択して、表示速度を優先にすることができます。この設定の効果は主にズーム機能を使用する際に感じることができます (ストラクチャーエディターのズーム参照)。

#### フォルダーが表示対象外のとき

このオプションはストラクチャーエディターで、フォルダーを使用した選択を使用した際の、テーブルが表示対象外のときのアピアランスを設定できます (フォルダーごとにテーブルをハイライト/薄暗くする参照)。**薄暗く表示** (テーブルイメージの代わりに影が表示される) または**非表示** (テーブルは完全に見えなくなる)が選択できます。

## フォームページ

このページでは4Dフォームエディターのデフォルトの動作と表示を設定できます。

### 移動

このオプショングループでは、フォームエディターでキーボードやマウスを使用してオブジェクトを移動する際のパラメターを設定できます。

#### キーボードを使用してステップ

このオプションでは**Shift**キーを押しながらキーボードを使用してオブジェクトを移動あるいはサイズ変更したときのステップ単位をポイントで指定できます。

#### ウィンドウの外に移動する場合

このオプションでは、フォームエディタにおいてウィンドウの境界外にマウスでオブジェクトを移動した際の動作を設定します。

- **自動スクロール**: このオプションが選択されていると、オブジェクトの移動に合わせてウィンドウ内でスクロールが行われます。この動作は大きなフォーム内でオブジェクトを移動する際に使用します。
- **ドラッグ＆ドロップを開始**: このオプションが選択されていると、エディタの外へのオブジェクトの移動はドラッグ＆ドロップとして解釈されます。フォームウィンドウは変更されず、移動されたオブジェクトは (例えば他のフォームなど) 他のウィンドウにドロップできます (アクションに互換性がある場合)。この動作は複数フォームでのオブジェクトのリサイクルやオブジェクトライブラリの利用時に使用します (カスタムオブジェクトライブラリの作成と使用参照)。

あなたの作業慣習や開発ニーズに応じてこのオプションを設定できます。

#### デフォルトで自動揃えを有効にする

このオプションはフォームエディターの新しいウィンドウ毎にデフォルトで自動揃えを有効にします。ウィンドウごとにこのオプションを変更できます (マグネティックグリッドを使用する参照)。

### デフォルト表示

このオプションを使用してフォームエディターの新規ウィンドウ毎にデフォルトで表示される項目を設定できます。フォームエディターの**表示**階層メニューを使用してウィンドウ毎に表示設定を変更できます。

- **デフォルトで表示するバッジ**: このオプションではフォームエディタの新しいウィンドウ毎に、どのバッジをデフォルトで表示するかを設定します。バッジに関する詳細はバッジを使用するを参照してください。
- **マーカーラインのカラー**: このオプションはフォームエディター中で (ヘッダー、ブレーク、詳細、フッター) エリアを定義するマーカーラインの色を設定します。マーカーに関する詳細は出力コントロールラインを使用するを参照してください。

# メソッドページ

このページではメソッドエディターのインターフェースやデフォルトの表示、および動作に関するオプションを設定します。ページは**スタイル**と**オプション**の2つのタブに分けられています。

## スタイル

### フォント

メソッドエディターの編集エリアのフォントと文字サイズを設定します:

- **フォント**: 使用するフォント名
- **サイズ**: 文字サイズ

オブジェクトのタイプごとに異なるフォントスタイルを設定できます。

### シンタックススタイル

このエリアのオプションを使用して、4Dのランゲージ項目 (フィールド、テーブル、変数、引数、SQL等) やSQLランゲージ項目 (キーワードや関数等) 毎に特定のスタイルやカラーを設定できます。コード保守のために、異なる色やスタイルの組み合わせを使用すると便利です。

- **ランゲージ**: 設定するランゲージをこのメニューを使用して**4D**または**SQL**から選択します。そしてリストされている項目についてメソッドエディター中で適用するスタイルを指定します。この項目リストはメニューで選択したランゲージに基づいて変わります。
- **項目タイプのリスト**: シンタックス項目の描画属性を設定するには表中で項目タイプごとにスタイルやカラーを設定します。項目ごとに複数のオプションを設定できます。項目ラベルは選択されたスタイルを反映して変更されます。カラーを変更するには項目のカラーアイコンをクリックして表示されるパレットで新しいカラーを選択します。
  **注:**
  - 4Dランゲージで、"プレーンテキスト"項目は他のタイプに属さないすべてのテキスト項目を指します (例: 記号、句読点、リテラル定数等)。
  - 4Dランゲージで、"キーワード"項目はマクロを使用してアクセスされるプログラム構造 (If/End if、Case of/End case等) を指します。SQLランゲージではすべてのコマンドとそれに割り当てられたキーワードを指します。

## オプション

### オプション

このエリアではメソッドエディターの表示オプションを設定します:

- **インデント**: メソッドエディター中での4Dコードで使用されるインデントの値を設定します。値はポイント (デフォルトで10) で設定します。
  4Dは構造を明確にするために自動でコードをインデントします:

```
1   If ($vlItemPos#0)
2      // Get the list item informatio
3     GET LIST ITEM(hlList;$vlItemPos;$v
4       // Is the item a Department iten
5   If ($vlItemRef ?? 31)
6      // If so, it is a double-click
7     ALERT("You double-clicked on the
8   Else
9      // If not, it is a double-click
10     // Using the parent item ID to
11    $vlDepartmentID:=List item parent
12    QUERY([Departments];[Departments]
13     // Tell where the Employee is w
14    ALERT("You double-clicked on the
15  End if
16 End if
17
18
```

  standard indentation

  アルゴリズムが複雑にになり、階層レベルが大きくなるような場合は、この値を減らすことで水平スクロールを減らすことができます。
- **行番号を表示**: メソッドエディターにデフォルトで行番号を表示するかどうかを設定します。メソッドエディターから直接カレントウィンドウの行番号を表示/非表示設定することもできます。
- **リスト表示**: オブジェクト (コマンド、テーブル、フィールド等) のリストを表示するかどうかを設定します。メソッドエディターから直接このリストを表示/非表示にすることができます。
- **括弧のマッチング**: コード中での対応する括弧のグラフィカルなシグナルを設定します。シグナルは括弧が選択されたときに表示されます。以下のオプションを選択できます:
  - **なし**: シグナルしません。
  - **四角**: 括弧が黒い四角で囲まれます。
  - **背景色**: 括弧がハイライトされます (色は後述の"カラー"エリアで設定します)。
  - **太字**: 括弧が太字で表示されます。

  デフォルトで四角オプションが選択されています:

```
INSERT MENU ITEM(main bar;-1;Get indexed string(79;1);FileMenu)
```

- **実行行をハイライト**: 通常の黄色の矢印インジケーターに加え、Duringで実行中の行をハイライトするかどうかを設定します。

```
    // Create a new empty
  hlList:=New list
    // Select all the rec
```

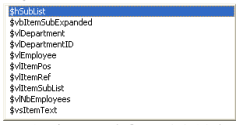
  このオプションの選択を解除すると、黄色の矢印のみが表示されます。

### 提案

このエリアではメソッドエディターの自動補完メカニズムを設定して、自分の作業習慣に合わせることができます。

- **ウィンドウを自動で開く**: 定数、ローカルおよびインタープロセス変数、テーブルの提案ウィンドウを自動で開くかを指定します。
  例えば"ローカルおよびインタープロセス変数"オプションにチェックされていると、$文字を入力したときに提案されるローカル変数のリストが表示されます:

```
GET LIST ITEM($
```




  対応するオプションのチェックを外すことで、要素ごとにこの機能を無効にできます。
- **提案の決定**: 自動補完ウィンドウに表示されたカレントの提案を受け入れる入力コンテキストを設定します。
  - **タブと区切り文字**: このオプションが選択されていると、タブキーまたは現在のコンテキストに関連する区切り文字で現在選択されている提案を決定することができます。例えば"ALE"と入力して"("を入力すると、4Dは自動で"ALERT("とエディターに書き込みます。区切り文字は以下の通りです: **( ; : = < [ {**
  - **タブのみ**: このオプションが選択されていると、現在の提案はタブキーを押したときにのみ受け入れられます。これは特に${1}のように、要素名に区切り文字を入力することを容易にします。.
    **注:** ウィンドウ中をダブルクリックするか、改行キーを押すことで提案を受け入れることもできます。

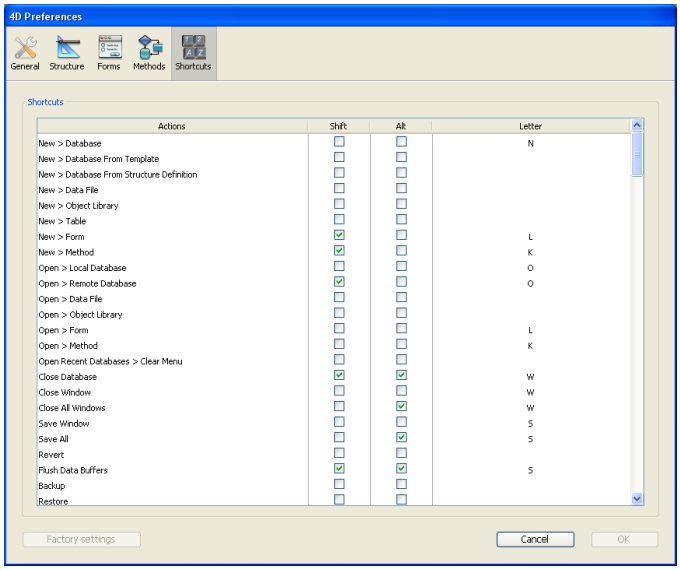
### カラー

このオプショングループでは、メソッドエディターインターフェースで使用されるカラーを設定します。

- **カーソルのある行 (アクティブウィンドウ) / カーソルのある行 (非アクティブウィンドウ)**: カーソルがある行の背景色。
- **検索された単語のハイライト**: 検索で見つかった単語のハイライト色。
- **括弧のハイライト**: 対応する括弧のハイライト色 (括弧のペアが色のハイライトで示されるときに使用されます、前述の"オプション"参照)。
- **編集エリアの背景色**: メソッドエディターウィンドウの背景色。
- **提案テキスト**: メソッドエディターで提案された自動補完テキストの色。

## ショートカットページ

このページには4Dのデザインモードで使用されるすべてのショートカットがリストされています (コピーなどのシステムショートカットを除く)。ショートカットを変更するには、設定する項目のモディファイアキー (ShiftとAlt) および文字をリスト中で変更します。またダブルクリックを行うと専用のダイアログボックスでショートカットの編集を行うことができます:




それぞれのショートカットには暗黙にCtrl (Windows) または Command (Mac OS) キーが含まれることに留意してください。

このリストは4D Extensions (4D v12) や Resources (v13) サブフォルダー内の*4DShortcutsvnn.xml*ファイルに基づいています。ダイアログボックスでショートカットをカスタマイズすると、このファイルがユーザーPreferencesフォルダーに複製され、デフォルトファイルの代わりに使用されます。そのため、4Dが更新されても、ショートカット環境設定は残されます。

# データベース設定


概要

ユーザー設定を使用する

一般ページ

インターフェースページ

コンパイラーページ

データベース/データストレージページ

データベース/メモリページ

移動ページ

バックアップ/スケジューラーページ

バックアップ/設定ページ

バックアップ/バックアップ＆復旧ページ

クライアント-サーバー/ネットワークオプションページ

クライアント-サーバー/IPオプションページ

Web/設定ページ

Web/オプション (I) ページ

Web/オプション(II)ページ

Web/ログ (タイプ) ページ

Web/ログ (バックアップ) ページ

Web/Web サービスページ

SQLページ

PHPページ

セキュリティページ

互換性ページ

## 概要

データベース設定では、カレントデータベースがどのように機能するかを設定します。これらのパラメターはデータベースに格納され、各データベースごとに異なる設定を保存できます。設定内容には待ち受けポート、デザインモードへのアクセス権、SQL設定などが含まれます。

ほとんどの設定は即座に有効になります。しかし (開始時モードの設定など) いくつかはデータベースが再起動されたときに有効となります。

**4D Server:** 2人以上のユーザーが同時にデータベース設定を変更しようとするとオブジェクトのロックが発生します。同時に1人のユーザーだけがデータベース設定ダイアログを使用できます。

**注**: 4Dは、4Dアプリケーションに適用できるもうひとつの設定グループを提供します。これは4Dアプリケーションレベルの設定です。詳細は環境設定を参照してください。

**データベース設定を開く**

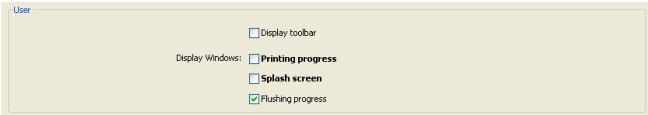
データベース設定ダイアログを開くには以下の手順を行います:

- **デザイン > データベース設定...** メニューオプションを使用する
- 4Dツールバーの設定ボタンをクリックする

- 4D Server上では、**編集** (Windows) または**ファイル** (MacOS) **> データベース設定...** メニューオプションを使用する

**パラメターのカスタマイズと初期設定**

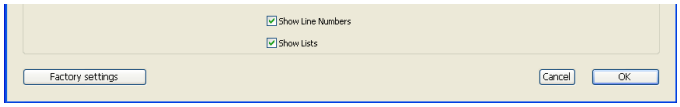
環境設定および設定ダイアログボックスでは、変更された設定内容は**太字**で表示されます:




環境設定内ではダイアログボックスで直接カスタマイズされるか、変換されたデータベースの場合以前のバージョンで変更された設定が、変更箇所として扱われます。

パラメターは手作業でデフォルト値に置き換えられたときにも太字で表示されます。このように、カスタマイズされたパラメターはすべて目視で識別することができます。

パラメターをデフォルト値に戻し、カスタマイズされたことを示す太字スタイルを取り除くためには、**初期設定**ボタンをクリックします:




このボタンをクリックすると、現在表示されているページのすべてのパラメターがリセットされます。このボタンは現在のページで最低一つのパラメターが変更されるとアクティブになります。

## ユーザー設定を使用する

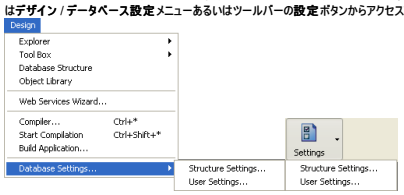
4D v13 ではデータベースのカスタム設定を外部ファイルとして生成できるようになりました。この機能が有効にされると、ユーザー設定を設定したときに、内容が外部ファイルに書き出されるようになります ("ユーザー設定")。そしてデータベースストラクチャーファイルに書き込まれたストラクチャー設定よりもこの設定が優先して使われます。

この結果 4D アプリケーションを更新しても、カスタム設定を保持できるようになります。あるいは異なる場所に展開する同じアプリケーションに対し、異なる設定を適用することが可能になります。またファイルの内容は XML で記述されるため、プログラムで設定を管理することができるようにもなります。

### "ユーザー設定"モードの有効化

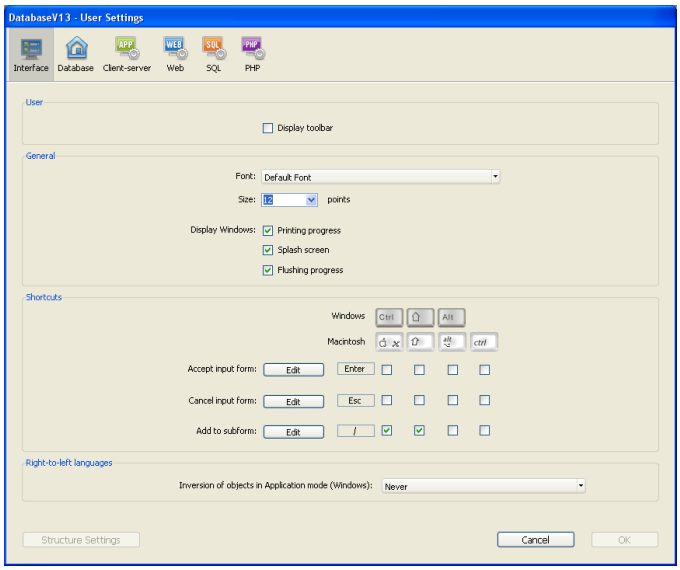
ユーザー設定の外部ファイル化を有効にするには、データベース設定のセキュリティページページ内で**外部ファイルのユーザー設定を有効にする**オプションを選択します。

このオプションをチェックすると、データベース設定が 2 つのダイアログ、**ストラクチャー設定**と**ユーザー設定**に分けられます。これらのダイアログボックスには**デザイン** / **データベース設定** メニューあるいはツールバーの**設定**ボタンからアクセスできます:




### ユーザー設定ダイアログボックス

外部保存モードが有効にされると、データベース設定は " ユーザー設定 "または " ストラクチャー設定 "、2 つのダイアログボックスから行えるようになります。
"ストラクチャー設定" ダイアログボックスは標準のデータベース設定と同じであり、すべてのプロパティにアクセスできます。"ユーザー設定"ダイアログボックスには外部ファイル化が可能なプロパティに関連するものが含まれます:




ユーザー設定が優先される原則に沿うために、このダイアログボックスを使用してオプションが変更されると、ストラクチャー設定に格納された値に代わりそちらが使用されます。

"ユーザー設定" ダイアログボックスに含まれるページのリストと、各ページでの標準設定との主な違いを以下の表にまとめます:

| データベース設定のページ | ユーザー設定のページ |
|---|---|
| 一般ページ | 利用できません |
| インターフェースページ | 標準設定と同じです |
| コンパイラーページ | 利用できません |
| データベース/データストレージページ | 利用できません |
| データベース/メモリページ | 標準設定と同じです |
| 移動ページ | 利用できません |
| バックアップ/スケジューラーページ | 利用できません |
| バックアップ/設定ページ | 利用できません |
| バックアップ/バックアップ＆復旧ページ | 利用できません |
| クライアント-サーバー/ネットワークオプションページ | 標準設定と同じです |
| クライアント-サーバー/IPオプションページ | 標準設定と同じです |
| Web/設定ページ | 標準設定と同じです |
| Web/オプション (I) ページ | 標準設定と同じです |
| Web/オプション(II)ページ | 標準設定と同じです |
| Web/ログ (タイプ) ページ | 標準設定と同じです |
| Web/ログ (バックアップ) ページ | 標準設定と同じです |
| Web/Web サービスページ | メソッドプリフィクスオプションが利用できません |
| SQLページ | 標準設定と同じです |
| PHPページ | 標準設定と同じです |
| セキュリティページ | 利用できません |
| 互換性ページ | 利用できません |

### ユーザー設定ファイル

データベース設定で**外部ファイルのユーザー設定を有効にする** オプションを選択し設定を行うと、ユーザー設定ファイルは以下の場所に作成されます :

```
[DatabaseFolder]/Preferences/settings.4DSettings
```

ここで [DatabaseFolder] はデータベースストラクチャーファイルが配置されたフォルダーの名前です。

このファイルの内容は XML であり、4D の XML コマンドや XML エディターを使用して読み込んだり変更したりできます。つまり、特に 4D Volume Desktop をマージしたエンジン組み込みアプリケーションにおいて、設定内容をプログラムで管理することが可能です。このファイルをプログラムで更新した場合、設定内容はデータベース再起動後に有効となります。

# 一般ページ

このページでは4Dデータベースの一般的なパラメターを設定します。

## デザイン

このエリアには**ツールバー表示**オプションがあります。このオプションがチェックされているとデザインモードで4Dツールバーが表示されます。

## 一般

ここではデータベースの開始時モードや表示にかかわるさまざまな設定を行います。

### 起動時モード

このメニューを使用して、データベースのデフォルトの起動モードを、**デザイン**モードまたは**アプリケーション**モードに設定できます。特記されない限り、パスワードアクセスシステムが有効になっていなければ、4Dはデフォルトでデザインモードで開きます。

**注:** データベースを開くダイアログオックスで、データベースをインタープリターモードで開くか、コンパイルモードで開くか選択できます (ローカルデータベースを開く参照)。

### 自動コメントを使用する

このオプションではデータベースで使用する自動コメントの有効/無効を切り替えられます。これらのパラメターについて詳細はコメントを使用するを参照してください。

**互換性に関するメモ**: 4D v12より、メソッドの先頭に書かれたコメントは、ヘルプTipとしてそのメソッド方のめっそどから参照された場合にメソッドエディター内で表示されます (WA SET EXTERNAL LINKS FILTERSの"ヘルプTipを使用する"参照)。この機能はメソッドのドキュメント化に有効であり、エクスプローラーに書かれたコメントと互換性がありません。ヘッダーエリアをドキュメントとして使用する場合、自動コメントは有効にしないでください。

## CPU優先度

CPU優先度エリアでは、4Dアプリケーションが他のアプリケーションに制御を戻す周期を変更することで、4Dアプリケーションのパフォーマンスを最適化できます。

実際4Dはマシンのオペレーティングシステム上で実行されていて、そのオペレーティングシステムは他のアプリケーションも実行しているかもしれません。4Dが実行されているシステムのアーキテクチャーはマルチタスク用に設計されていないため、システムはアプリケーションに制御を渡し、アプリケーションから制御が返されたら次のアプリケーションに制御を渡します。アプリケーションがシステムに制御を戻すとき、他のアプリケーションに制御を渡す時間制限の希望値を指定します。

動作環境や特定のニーズに基づき、同じマシンで動作する他のアプリケーションに対する4Dの優先度を設定することができます。同じデータベースに対し、優先度は4D (ローカルモード)、4D Server、そして4Dリモートモードと個別に設定できます。これらの設定はストラクチャーファイルに保存されます。

デフォルトでは平均的な値が設定されています (カーソルがサーモメーターの中央にあります)。デフォルトではマシンタイムの約20%を消費するようになっています。時にはこの設定値が適切でない場合があります。例えば4D Server専用のマシンを用意する場合、マシンタイムの80%をほかに何もすることのないシステムに使用させるのはもったいありません。この場合、"4D Server"の優先度を上げることでパフォーマンスを向上できます。逆に他のケースで、4Dの優先度を下げ、他のアプリケーションにマシンタイムを渡したほうがよい場合もあります。

**注**

- Get database parameterとSET DATABASE PARAMETERコマンドを使用して、それぞれの値をプログラムで設定できます。
- **警告:** これらの値を不適切に設定するとアプリケーションのパフォーマンスが悪化します。

# インターフェースページ

インターフェースページでは、データベースのインターフェースに関するさまざまな設定を行います。

**一般**

このエリアではデータベースの表示に関する様々なオプションを設定するために使用します。

**フォントとサイズ**

**選択...** ボタンを使用して、MESSAGE コマンドで使用される文字のフォントとサイズを設定します。

デフォルトのフォントとサイズは、4Dが開かれるプラットフォームによって異なります。

**注:** このプロパティは4Dの以下の部分にも影響します:

- エクスプローラー のプレビューの特定のエリア
- フォームエディター のルーラー

他のオプションではアプリケーションモードでのさまざまなウィンドウの表示を設定します。

- **印刷の進捗状況**: 印刷時の印刷進捗状況を表示するダイアログを有効/無効にします。
- **スプラッシュスクリーン**: このオプションが選択解除されていると、アプリケーションモードで、カレントメニューバーのスプラッシュスクリーンは表示されません (メニューバーの管理参照)。このウィンドウを非表示にした場合、ウィンドウの表示は例えばOn Startupデータベースメソッドの中からプログラムで管理しなければなりません。
- **フラッシュの進捗状況**:このオプションがチェックされていると、キャッシュ中のデータがフラッシュされる際、4Dは画面左下にウィンドウを表示します。この処理はユーザーの操作をブロックするため、ウィンドウを表示することでフラッシュが行われていることをユーザーに通知することができます :

  **注:** データベース設定のデータベース/メモリページでキャッシュフラッシュの周期を設定できます。

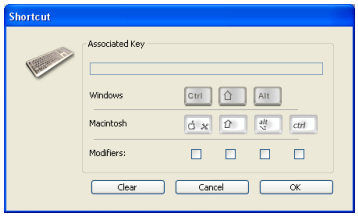
**ショートカット**

ショートカットエリアでは、4Dの基本的な3つのデフォルトショートカット (入力フォーム確定、入力フォームキャンセル、サブフォームに追加) を確認し、変更することができます。これらのショートカットは両プラットフォームで同じです。キーの形をしたアイコンはWindowsとMac OSそれぞれのキーの組み合わせを示しています。

デフォルトのショートカットは以下のとおりです:

- 入力フォーム確定: **Enter**
- 入力フォームキャンセル: **Esc**
- サブフォームに追加: **Ctrl+Shift+/** (Windows) または **Command+Shift+/** (Mac OS)

操作のショートカットを変更するには、それぞれの**編集**ボタンをクリックします。以下のダイアログが表示されます:




ショートカットを変更するにはキーボードで新しいキーの組み合わせをタイプし、**OK**をクリックします。ショートカットを無効にしたい場合**クリア**ボタンをクリックします。

**Right-to-left言語**

"right-to-left言語"という用語はアラビア語など、右から左に向かって読む言語を指しています。

4DはWindowsにおいてright-to-left言語を拡張サポートしています。"right-to-left"モードを有効にすると、アプリケーションモードで、4Dデータベースのインターフェース (フォームやメニュー) は自動で反転されます。これはWindowsのみで有効です。Mac OS Xでは同等の機能はありません。

この反転はテキストオブジェクトを読む方向だけでなく、グラフィックオブジェクトやフォーム中のオブジェクトの位置、ウィンドウタイトルやボタン、メニューの位置等に影響します。

**注:** 現時点でフォームの印刷時には、反転モードは使用できません。

**アプリケーションモードでオブジェクトを反転**オプションを使用して、データベースレベルで"right-to-left"モードを有効にし設定できます。データベースが実行されるコンテキストに基づき、このモードを設定する3つのオプションを使用できます:

- **しない**: このオプションが選択されていると、OSが"right-to-left"モードに設定されていても、データベースがこのモードに変更されることはありません。デフォルトでこのオプションが選択されます。
- **自動**: このオプションが選択されていると、OSが"right-to-left"モードに設定されている場合、データベースもこのモードに変更されます (例えばアラビア語バージョンのWindows)。
- **常に**: このオプションが選択されていると、OSの設定が何であれ、データベースは"right-to-left"モードに変更されます。

このオプションを変更すると、設定を保存すると即座にデータベースに適用されます。

**注:**

- このグローバルな設定はSET DATABASE PARAMETERコマンドでプログラムすることもできます。
- right-to-leftモードが有効の場合でも、各フォームごとにそれを無効にできます (フォームプロパティ の"オブジェクトを反転しない (Windows)"を参照)。
- デフォルトでright-to-leftモードはフォーム中のピクチャーを反転しません。必要であれば追加のオプションを使用してピクチャーを反転できます (スタティックピクチャーを使用するの"ピクチャーのミラー効果 (Windows)"参照)。

# コンパイラーページ

このページではアプリケーションのコンパイルに関連するパラメーターを設定できます。コンパイルについてはコンパイルを参照してください。

**コンパイルオプション**

このエリアにはコンパイル処理中に使用される一般的なオプションがグループ化されています。

- **範囲チェック**: 範囲チェックの有効/無効を設定します。範囲チェックは追加の検証であり、実行時のアプリケーションオブジェクトの状態に基づき、コードをチェックします。範囲チェックに関する詳細は範囲チェックを参照してください。
- **Symbolファイルを生成**: 変数とその型のリスト、およびメソッドの引数や戻り値の型のリストが記述されたドキュメントを生成するために使用します。
  ファイルはデータベースストラクチャーと同階層に*DatabaseName_symbols.txt*という名前で作成されます。
  シンボルファイルに関する詳細はシンボルファイルを参照してください。
- **エラーファイルを生成**: シンタックスチェック時にエラーファイルを生成するために使用します。一般的なエラー、特定のコード行のエラー、および警告などが含まれます。
  コンパイラーが検知したすべてのエラーは4Dのメソッドメニューからアクセスできます。エラーファイルを生成し、他のマシンに転送することが有効な場合もあります。特に複数の開発者がチームで開発を行っている場合などです。
  エラーファイルはXML形式で生成されるため、内容を自動で解析することが容易になっています。このファイルを使用してカスタマイズされたエラー表示インターフェースを作成することも可能です。エラーファイルは*DatabaseName_errors.xml*という名称で、ストr貫う茶ーファイルと同階層に作成されます。
  エラーファイルに関する詳細はエラーファイルを参照してください。
- **64-bit プロセッサー用にもコンパイルする**: アプリケーションを (32-bitコンパイルに加え) 4D 64-bit版用にコンパイルするために使用します。以下の点に留意してください:
  - バージョン12.1より64-bitバージョンの4D Serverが提供されています。
  - 非Unicodeモードのアプリケーションを64-bitにコンパイルすることはできません。
  - 64-bit版の4Dでアプリケーションを使用する場合にこのオプションを使用します。OSが64-bitであるかどうかは関係ありません。
- **ローカル変数初期化**: メソッドの先頭でのローカル変数初期化モードを設定するために使用します:
  - **ゼロにする**: 変数はデフォルトで0に設定されます (文字型の場合空の文字列、数値型の場合0等)
  - **ランダム値にする**: コンパイラーは無意味な値を割り当てます。この値は常に同じであり、例えば倍長整数の場合1919382119、文字の場合"rgrg"、ブールの場合True等になります。このオプションを使用すると、初期化を忘れているローカル変数を探すのが容易になります。
  - **いいえ**: コンパイラーはローカル変数を初期化しません。この設定を使用するとアプリケーションの実行時間を短縮できます。ただし開発者が正しくローカル変数を初期化しなければなりません。
- **コンパイルパス**: コンパイラーの実行周期数を設定するために使用します。
  - **すべて定義させる**: コンパイルを行う際に可能なステージをすべて行います。
  - **ローカル変数のみ自動定義させる**: プロセスおよびインタープロセス変数の型を決定する処理は行われません。このオプションを選択する場合、すべてのプロセス変数とインタープロセス変数を開発者自身が宣言するか、自動コンパイラーメソッド生成機能を使用しなければなりません。
  - **自動変数定義は行わない**: ローカル、プロセスおよびインタープロセス変数の型を決定する処理は行われません。このオプションを千tかうする場合、すべての変数が明示的に宣言されていなければなりません。

**デフォルトの型指定**

このエリアでは曖昧なアプリケーションオブジェクトのデフォルト型を設定します。

- **数値**: 実数または倍長整数から選択できます。アプリケーション中に型指定ディレクティブが書かれている場合、そちらが優先されます。倍長整数を指定することでアプリケーションを最適化できます。
- **ボタン**: 実数または倍長整数から選択できます。アプリケーション中に型指定ディレクティブが書かれている場合、そちらが優先されます。標準ボタンの他以下のオブジェクトが対象となります: チェックボックス、3Dチェックボックス、ハイライトボタン、非表示ボタン、3Dボタン、ピクチャーボタン、ボタングリッド、ラジオボタン、3Dラジオボタン、ピクチャーラジオボタン、ピクチャーポップアップメニュー、階層ポップアップメニュー、ポップアップ/ドロップダウンリスト。

**コンパイラーメソッド...**

このエリアでは、コンパイラーが自動で生成するコンパイラーメソッドの名前を設定できます。これらのメソッドはすべての変数の型定義をグループ化 (プロセス変数、インタープロセス変数、プロセス配列、インタープロセス配列、メソッド) します。これらのメソッドを生成するには、コンパイラーウィンドウの変数定義を生成ボタンを使用します。詳細はコンパイラーウィンドウを参照してください。

最大5つのコンパイラーメソッドが生成されます。コンパイラーメソッドはアプリケーションに対応するオブジェクトが存在する場合のみ作成されます:

- **変数**: プロセス変数定義がグループ化されます。
- **インタープロセス変数**: インタープロセス変数定義がグループ化されます。
- **配列**: プロセス配列定義がグループ化されます。
- **インタープロセス配列**: インタープロセス配列定義がグループ化されます。
- **メソッド**: メソッド引数として使用されるローカル変数定義がグループ化されます (例: C_INTEGER(mymethod;$1))。

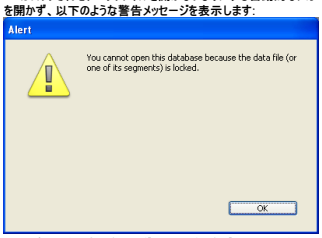
それぞれの対応するエリアで、作成されるメソッドの名前を編集できます。
常に"Compiler_"という接頭辞が付きます。これは変更できません。各メソッド名はユニークでなければならず、接頭辞を含み31文字以下でなければなりません。拡張文字 (アクセント文字や記号文字等) やスペースは使用できません。

# データベース/データストレージページ

このページでは4Dデータベースが使用するデータストレージに関する設定を行います。

**一般設定**

- **ストラクチャーファイルが読み込みのみのときに警告を表示しない**: ストラクチャーファイルが読み込みのみモードのデータベースを開く際、4Dはその旨を通知するダイアログを表示します。そのデータベースを開いた場合、ストラクチャーに対する変更は保存されません。
CD-ROMで配布されるデータベースなど、特定のケースではこの警告を表示させたくない場合があります。その場合このオプションをチェックしてください。
- **読み込みのみのデータファイルの使用を許可する**: このオプションを使用すると、OSレベルでロックされているデータファイルをアプリケーションから開けるようになります。
4Dはロックされたデータファイルを開かないようにする自動的なメカニズムを実装しています。データファイルがロックされていると、4Dはデータベースを開かず、以下のような警告メッセージを表示します:



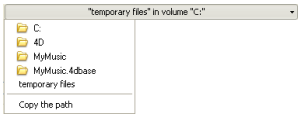

このオプションを選択しないかぎり、ロックされたデータファイルを開くことはできません (4Dデータベースのデフォルト動作)。

***ロックされたファイルについて**: ロックされたファイルは読むことはできても内容を変更することはできません。例えば (DVDのような) 編集不可のメディアに格納されたファイルや、このようなメディアからコピーされたファイルは、ロックされていることがあります。DVDに格納されたデータベースを使用するような場合、ロックされたデータファイルを使用できるようアプリケーションを設定することができます。*
*しかしこの場合、データを追加したり変更したりできない点に注意してください。不注意にこのような状況にならないよう、4Dではデフォルトでロックされたデータファイルを開くことを禁止しています。*

**テンポラリファイルの場所**

このエリアでは、4D実行中に作成されるテンポラリファイルの場所を設定できます。テンポラリファイルフォルダーは必要に応じて、一時的にメモリ中のデータをディスクに保存するためにアプリケーションが使用します。

現在のフォルダーの場所は"現在:"エリアに表示されます。このエリアをクリックするとパス名がスクロールダウンリスト形式で表示されます:




3つのオプションが提供されます:

- **システム**: このオプションが選択されると、4Dテンポラリファイルは OSが指定する場所に配置されたフォルダーに作成されます。使用するシステムが指定する場所はTemporary folderコマンドで知ることができます。ファイルはデータベース名とユニークな識別子からなるサブフォルダーに置かれます。
- **データファイルフォルダー** (デフォルト): このオプションが選択されると、4Dテンポラリファイルはデータベースのデータファイルと同階層に配置される"temporary files"フォルダーに作成されます。
- **ユーザー指定**: このオプションは場所をカスタマイズするために使用します。

場所のオプションを変更したら、設定を有効にするためにアプリケーションを再起動する必要があります。4Dは選択されたフォルダーに書き込みアクセス権があるかを確認します。アクセス権がなければ、4Dは使用できるフォルダーが見つかるまで他のオプションを試します。

**注:** このオプションは、ストラクチャー定義がXML形式で書き出されたとき、ストラクチャーの"extra properties"に格納されます (ストラクチャー定義の書き出しと読み込み参照)。

**テキスト比較**

これらのオプションのいずれかを変更した場合、新しい設定を有効にするためにアプリケーションを終了し、再起動しなければなりません。データベースが再び開かれると、すべてのインデックスが自動で再作成されます。

- **@がテキストパターンの最初または最後にある場合のみ、ワイルドカードとして扱う**: クエリ条件や文字列比較の際に"@"をどのように解釈するかを設定します。
このオプションが選択されていない場合 (デフォルト設定)、@はワイルドカードとして扱われ、あらゆる文字の代わりとみなされます (ワイルドカード記号 (@)参照)。
このオプションが選択されている場合、@が単語中にある場合に普通の文字として扱われます。この設定は電子メールアドレスの検索などに有効です。このオプションは検索、並び替え、文字列比較、およびテーブルに格納されるデータや配列などメモリ中のデータに影響を与えます。(インデックス付きか否かにかかわらず) 文字型やテキスト型のフィールドおよび変数が、検索や並び替え時に@文字がどのように解釈されるかの影響を受けます。
**注:**
  - 検索の場合、@が条件の先頭か最後にある場合、"@"がワイルドカードとして扱われることに注意することが重要です。単語の中に"@"文字がある場合のみ (例: bill@cgi.com)、4Dは異なった扱いをします。
  - このオプションはobject引数に"@"ワイルドキャラクターを受け入れるオブジェクト(フォーム)テーマのコマンドの動作にも影響を与えます。
  - セキュリティの理由で、データベースのAdministratorまたはDesignerのみがこのパラメーターを変更できます。
- **現在のデータ言語**: 文字列の処理と比較に使用する言語を設定します。言語の選択は直接的にテキストの検索や並び替え、および文字種の判定に影響を与えます。しかし翻訳テキストや日付、時間、通貨単位などには影響を与えず、システム言語が使用されます。デフォルトで4Dはシステム言語を使用します。
この設定を変更することで4Dデータベースはシステム言語とは異なる言語で動作することができます。データベースが開かれると、4Dエンジンはデータファイルが使用する言語を検知し、ランゲージに通知します (インタープリターおよびコンパイルモード)。データベースエンジンあるいはランゲージいずれで実行されるにしても、テキスト比較は同じ言語で実行されます。
**注:** 4D環境設定でも言語を設定できます (一般ページ参照)。この場合その設定は新規に作成されるデータベースに適用されます。
- **非文字・非数字のみをキーワード区切り文字とする**: 4Dが使用するキーワード区切り文字の設定を変更し、その結果、作成されるキーワードインデックスに影響を与えます。このオプションが選択されていない場合、4Dは言語上の特質を考慮する洗練されたアルゴリズムを使用します。このアルゴリズムは、文字列中をダブルクリックされたときに選択範囲を決定するためワープロソフトが使用するものと同じです。このアルゴリズムに関する詳細は以下のWebページを参照してください: http://userguide.icu-project.org/boundaryanalysis
このオプションが選択されている場合、4Dは簡易的なアルゴリズムを使用します。この設定では文字でも数字でもない文字がキーワード区切り文字として扱われます。この設定は日本語など特定の言語の要求に沿うものです。
- **旧バージョン互換の文字列比較を使用する**: このオプションはデータ言語に日本語が選択されている場合にのみ表示され、4Dが文字列比較の際に使用するアルゴリズムのパラメーターを変更するために使用されます。この設定は文字列の検索や並び替え、および文字列の比較に影響します。
このオプションが選択されない場合、デフォルトのパラメーターを使用した文字列比較アルゴリズムが使用されます。結果、例えば長音記号は直前の文字の繰り返し文字として解釈されます。直前に文字がない場合は記号として解釈されます。これはUnicodeのデフォルト設定ですが、4D2004以前とは異なる動作です。
このオプションを選択することで、検索や並び替え、文字列比較の結果が4D2004と同じになります。例えば長音記号は直前に文字があるかないかにかかわらず、常に記号として扱われます。
このオプションはICUのコレーションオプションの"Traditional Sorting Order"に対応します。

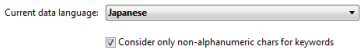
**Mecabのサポート(日本語版)**

日本語版の4Dでは、*Mecab* ライブラリーがサポートされており、日本語用に調整されたキーワードのインデックスアルゴリズムを兼ね備えています。

v14以降の日本語版の4Dではこの新しいアルゴリズムがデフォルトで使用されています。*Mecab* ライブラリーを使用するために必要なファイルは全て4Dアプリケーションの **Resources** フォルダ内の **mecab** フォルダー内にインストールされています(日本語版のみ)。

必要であれば *Mecab* アルゴリズムを無効にして、以前と同じ *ICU* ライブラリーを使用することもできます。

*Mecab* を無効化するためには、データベース設定内の**非文字・非数字のみをキーワード区切り文字とする**のオプションにチェックを入れるだけです:




**注:** 日本語版4Dの、**Resources/mecab** folderフォルダを削除または改名することによっても *Mecab* を無効化することができます。

## データベース/メモリページ

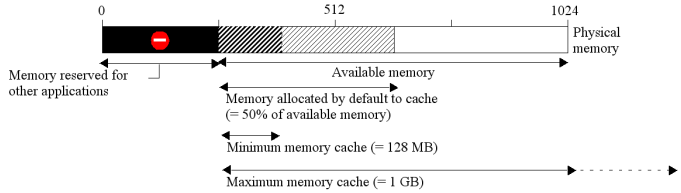
このページではデータベースのキャッシュメモリに関する設定を行います。

**データベースキャッシュ設定**

- **起動時にキャッシュサイズを計算**: このオプションが選択されている場合、キャッシュメモリの管理は、設定された制限に基づき、4D起動時にキャッシュサイズが計算されます。これによりほとんどのケースで高パフォーマンスのメモリ設定が行われます。キャッシュメモリのサイズは設定されたパラメターに基づき動的に計算されます。デフォルトで提案される値は標準の4Dの使用状況に対応します。

  - **他のアプリケーションとシステムのために予約するメモリ**: システムや他のアプリケーションが使用するために取り置くRAMメモリ量。4Dが実行されるマシン上で他のアプリケーションも実行する場合、必要に応じてこの値を増やします。
  - **利用可能なメモリからキャッシュに使用する率**: 残りのメモリからキャッシュに割り当てる量の率。
    デフォルトでキャッシュに割り当てるサイズを取得するためには、以下の計算式を適用します: (物理メモリ – 予約したメモリ) X キャッシュに使用するメモリのパーセンテージ。動的に計算するモードの場合、キャッシュメモリのサイズはアプリケーションやシステムのニーズに応じて動的に変化します。以下のオプションを使用して上限と下限を設定できます:
  - **最小サイズ**: キャッシュ用に予約するメモリの最小量。この値は4MB以上でなければなりません。
  - **最大サイズ**: キャッシュが使用することのできるメモリの最大量。32 bitバージョンの4Dおよび4D Serverでは、パフォーマンスの理由で、内部的にこの値は 2000 MB (2 GB) に制限されています。64 bitバージョンの4D Serverでは制限はありません。

  制限の設定は、メモリ搭載量が不明であるマシン用にアプリケーションを配布する際に便利です。この場合、制限を設定することで最低限のパフォーマンスを保証できます。この動作を図示すると以下のようになります:
  キャッシュメモリの計算例:
  *予約するメモリ量 = 256 MB*
  *キャッシュに利用するパーセンテージ = 50%*
  *最大サイズ = 1 GB 最小サイズ = 128 MB*




  **動的計算モードの設定例**
  ここではいくつかの例を使用して、さまざまなソフトウェアとメモリ量に基づき動的な計算がどのように行われるか見ていきます。
  - 768 MBのRAMが搭載され4D Serverが1つだけ実行されるマシン。この場合、最大限キャッシュメモリを割り当てることを考慮できます:
    - 予約するメモリ量 = 256 MB.
    - キャッシュに利用するパーセンテージ = 100% 1つしか4D Serverアプリケーションがないため
    - 最小サイズ = 10 MB、最大サイズ = 1GB (この例ではこれらの値は意味を持ちません)

    --> デフォルトキャッシュサイズ: (768 - 256) x 100% = 512 MB
  - 同様のマシンで3つの4D Serverを動作させる場合。それぞれのアプリケーションに同じキャッシュサイズを割り当てるにはそれぞれのアプリケーションに33%づつキャッシュを割り当てます。
    --> 各アプリケーションのデフォルトキャッシュサイズ: (768 - 256) x 33% = 170 MB
  - アプリケーションを広く配布する場合、メモリがどのように実装されているかは不明です。テストの結果キャッシュサイズが20〜100 MBのときにベストなパフォーマンスが出ることを確認しました。
    - 予約するメモリ量 = 256 MB.
    - キャッシュに利用するパーセンテージ = 50% (他のアプリケーションのために利用可能なメモリを残しておきます)
    - 最小サイズ = 20 MB、最大サイズ = 100 MB.

    --> 256 MBのRAMを実装したマシンの場合、4Dは最小 = 20 MBのキャッシュサイズを使用します。
    --> 1 GBのRAMを実装したマシンの場合、4Dは最大 = 100 MBのキャッシュサイズを使用します。
    --> 384 MBのRAMを実装したマシンの場合、4Dは (384 - 256) x 50% = 64 MBのキャッシュサイズを使用します。
- **動的キャッシュの計算がチェックされていない場合**: このモードではデータベースのキャッシュメモリサイズを開発者が決定します。4Dはキャッシュメモリを設定する入力エリアと、物理メモリに関する情報 (マシンに実装されたRAM)、現在のキャッシュ、そして再起動後のキャッシュサイズを表示します。

  入力されたキャッシュサイズは、マシンのリソース状況にかかわらず、4Dデータベース用に予約されます。この設定はメモリ搭載量が分かっているなど特別な状況で使用できます。ほとんどのケースで動的計算モードのキャッシュで良いパフォーマンスを提供します。

- **4D Serverと4D Volume Desktopでキャッシュを物理メモリに保持する**
  この上級オプションは、キャッシュをマシンの物理メモリ (RAM) に強制的に保持させるために使用します。これはデータベースが4D Volume Desktopまたは4D Serverで実行されている場合に有効です。
  デフォルトでは必要に応じて、キャッシュの一部または全体はマシンの仮想メモリにフラッシュされることがあります (仮想メモリはハードディスク上に確保されるエリアです)。特定のケースでは、このメカニズムにより、4D Serverや4D Volume Desktopで配布されたアプリケーションにパフォーマンスの低下が見られるかもしれません。その場合、このオプションを選択してキャッシュが物理メモリ上に保持されるよう設定できます。もちろんマシン設定 (RAMのサイズ、同時に実行されるアプリケーションの数等) がこの原則に適応しなくてはなりません。
  **注:** このオプションは4Dで実行されるアプリケーションでは効果がありません。

- **データをディスクに保存 …秒/分**: キャッシュ中のデータを自動的に保存する間隔を設定します。
  4Dはキャッシュ中のデータを定期的に保存します。この間隔を1秒から500分の範囲で設定できます。デフォルトの設定値は20秒です。この保存はキャッシュがいっぱいになった場合やアプリケーションを終了する際にも行われます。
  大量のデータ入力が予期される場合は、この間隔を短くすることを検討してください。停電などの理由でマシンが停止すると、前回の保存以降に入力されたデータが失われてしまいます (データログファイルをとっていればそれから復旧させることは可能です)。
  キャッシュがフラッシュされるたびにデータベースの動作が遅くなる場合、周期を調整する必要があります。動作が遅くなるのは、大量のレコードがディスクにフラッシュされるためです。フラッシュ周期を短くすることで各フラッシュ時に保存されるレコード数を減らすことができ、動作も速くなります。
  デフォルトで4Dはキャッシュがフラッシュされていることを示す小さなウィンドウを表示します。このウィンドウを表示したくない場合、インターフェースページページの**フラッシュの進捗状況**オプションの選択を解除します。

## 移動ページ

データベース設定の移動ページを使用して、デザインモードでオブジェクトを移動した場合のデフォルト動作を設定できます。4Dはこのデータベース (移動先データベース) にドロップ/ペーストされたオブジェクトに対しこれらの設定を適用します。

**依存するオブジェクトをコピーする際のデフォルトアクション**

これらのオプションは依存オブジェクト、言い換えれば移動されるフォーム (CHANGE LICENSES参照) にリンクされたオブジェクトの移動を設定します。

これらのデフォルトアクションは、オブジェクトの移動時にコンフリクトが発生せず、**名前の衝突時のみ**が移動ダイアログを表示で選択されている場合 (次節参照)、自動で適用されます。そうでない場合、移動ダイアログが表示され、その中でデフォルトの選択肢として使用されます。

4Dは各オブジェクトごとに**無視**、**作成 (必要なら名前を変更)**、**作成 (必要なら置換)**、そして**他のオブジェクトを使用**オプションを提供します。移動ダイアログボックスが表示される場合、オブジェクトごとにより詳細なオプションが提供されることに留意してください。

- **無視**: この場合依存オブジェクトは移動先データベースにコピーされません。移動ダイアログにおいてデフォルトで**作成しない**アクションが選択されます。
- **作成 (必要なら名前を変更)**: この場合依存オブジェクトは常に移動先データベースにコピーされます。オブジェクトが移動先データベースに存在しない場合、移動ダイアログにおいてデフォルトで**作成**アクションが選択されます。
  移動先オブジェクトにすでにその名前を持つオブジェクトが存在する場合、コピーされるオブジェクトの名前には接尾辞"_X"が追加されます。この場合、移動ダイアログにおいてデフォルトで**名称変更**アクションが選択されます。
- **作成 (必要なら置換)**: この場合依存オブジェクトは常に移動先データベースにコピーされます。オブジェクトが移動先データベースに存在しない場合、移動ダイアログにおいてデフォルトで**作成**アクションが選択されます。
  移動先オブジェクトにすでにその名前を持つオブジェクトが存在する場合、既存のオブジェクトはコピーされるオブジェクトで上書きされます。この場合、移動ダイアログにおいてデフォルトで**置換**アクションが選択されます。
- **他のオブジェクトを使用**: このオプションを選択すると、名前の衝突時のみが選択されていても、システマチックに移動ダイアログが表示されます。オブジェクトを移動する際、コピーされる依存オブジェクトの代わりに使用する移動先データベースのオブジェクトを選択しなければなりません。

**注:** これらのオプションは依存オブジェクトのみを対象とします。移動されるオブジェクト自身のデフォルトアクションは**作成 (必要なら名前を変更)**です。

- **移動ダイアログを表示**: このメニューでは移動ダイアログボックスの表示を設定します。
  **常に**オプションが選択されていると、オブジェクトが移動されるたびに移動ダイアログボックスが表示されます。これにより移動操作をより詳細に設定できます。**名前の衝突時のみ**オプションを選択すると、移動される (依存またはメインの) オブジェクトが移動先データベースのオブジェクトと名前の衝突を起こしている場合のみ、ダイアログが表示されます。

## バックアップ/スケジューラーページ

このページにあるオプションを使用して、データベースの自動バックアップのスケジュールを設定できます。標準のクイック設定、または完全なカスタマイズを選択できます。

**自動バックアップ**メニューでの選択に基づき、さまざまなオプションが表示されます:

- **しない**: スケジュールに基づくバックアップは無効となります。
- **毎時**: 次の時間以降、毎時間ごとに自動バックアップを行います。
- **毎日**: 日に一回自動バックアップを行います。バックアップを何時に行うかを設定します。
- **毎週**: 週に一回自動バックアップを行います。バックアップを開始する曜日と時刻を入力するエリアが表示されます。
- **毎月**: 月に一回自動バックアップを行います。バックアップを開始する日付と時刻を入力するエリアが表示されます。
- **個別**: 自動バックアップを詳細にスケジュールする場合に使用します。このオプションを選択すると、複数の入力エリアが表示されます:
  - **時間ごと**: 時間単位でバックアップの間隔をスケジュールできます。1から24までの値を設定できます。
  - **日ごと**: 日単位でバックアップの間隔をスケジュールできます。例えば1を設定すると毎日バックアップが行われます。このオプションを選択した場合、バックアップが開始される時刻を設定しなければなりません。
  - **週ごと**: 週単位でバックアップの間隔をスケジュールできます。例えば1を設定すると毎週バックアップが行われます。このオプションを選択した場合、バックアップを開始する曜日と時刻を設定しなければなりません。複数の曜日を選択できます。例えば毎週水曜日と金曜日にバックアップをするようプログラムできます。
  - **月ごと**: 月単位でバックアップの間隔をスケジュールできます。例えば1を設定すると毎月バックアップが行われます。このオプションを選択した場合、バックアップを開始する日付と時刻を設定しなければなりません。

#### 4Dのバックアップに関する留意点

4Dのバックアップメカニズムについてはデータベースのバックアップと復元を参照してください。

クライアント/サーバーモードでは、バックアップパラメターは4D Serverマシンでのみ設定できます。

##### Backup.XMLファイルを使用した設定の管理

すべてのデータベースバックアップおよび復旧の設定はXMLファイルに書き出されます。4Dはデータベース設定ダイアログボックスにオプションを表示するためと各バックアップが開始される際に、このファイル内のデータを使用します。

4DのXMLコマンドやXMLエディターを使用して設定内容を取得したり変更したりできます。これにより開発者は、特に4D Volume Desktopでビルドされたアプリケーションにおいて、バックアップ設定をプログラムで管理できます。

XML形式のバックアップおよび復旧設定ファイルはBackup.XMLという名前で、自動で以下の場所に作成されます:
*DatabaseFolder/Preferences/Backup/Backup.XML*
*... DatabaseFolder* はストラクチャーファイルが格納されているフォルダーです。サブフォルダー*Preferences/Backup/* はなければ自動で作成されます。バックアップの環境設定に使用されるXMLキーは*XML Keys-Backup*マニュアルで説明されています。

## バックアップ/設定ページ

このオプションのページを使用して、バックアップするファイルやその場所、およびログファイルを設定します。

### 内容

このエリアでは次回のバックアップ時にコピー対象とするファイルやフォルダーを指定します。

- **データファイル**: データベースデータファイル。このオプションが選択されている場合、(存在すれば) データベースのカレントログファイルがデータとともにバックアップされます。
- **ストラクチャーファイル**: データベースストラクチャーファイル。コンパイルされ4D Volume Desktopが結合されたデータベースの場合、このオプションを選択するとWindowsでは.exeファイルが、Mac OSではパッケージがバックアップされます。
- **ユーザーストラクチャーファイル** (任意): カスタマイズされたユーザーフォームが格納されたユーザーストラクチャーファイル (作成されている場合)。詳細はユーザーフォームを参照してください。
- **添付**: このエリアではデータベースと同時にバックアップの対象とするファイルやフォルダーを指定します。ここではどのようなタイプのファイル (ドキュメントやプラグイン、テンプレート、ラベル、レポート、ピクチャーなど) でも指定できます。
  ファイルまたはその内容をバックアップするフォルダーを個々に設定できます。添付エリアには設定されたファイルのパスが表示されます。
  - **削除**: 選択したファイルを添付エリアから取り除きます。
  - **フォルダー追加...**: バックアップに追加するフォルダーを選択するダイアログボックスを表示します。復元の場合、フォルダーがその内容物とともに復元されます。データベースファイルを含むフォルダーを除き、すべてのフォルダーやマシンに接続されたボリュームを選択できます。
  - **ファイル追加...**: バックアップに追加するファイルを選択するダイアログボックスを表示します。

4Dデータベースファイルについては4Dファイルの説明を参照してください。

### バックアップファイル保存先

このエリアではバックアップファイルの格納場所を確認したり、変更したりできます。

エリアをクリックすると、ファイルの場所がポップアップで表示されます。

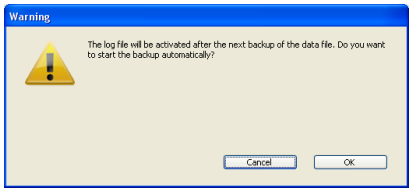
バックアップファイルの格納場所を変更するには、[...]ボタンをクリックします。選択ダイアログが表示され、バックアップファイルを配置するフォルダーやディスクを選択できます。"使用状況"と"空き容量"エリアは、選択したフォルダーが存在するディスクの状態を自動で表示します。

### ログ管理

**ログファイルを使用**オプションが選択されていると、データベースはログファイルを使用します。ログファイルの場所はオプションの下に表示されます。このオプションが選択されている場合、ログファイルなしでデータベースを開くことはできません。

4Dで作成されたデータベースはデフォルトでログファイルを使用します。このファイルの作成は自動バックアップの計画に含まれています (自動バックアップ参照)。ログファイルには*DataFileName.journal*のように名前が付けられ、データベースストラクチャーと同階層に置かれます。

新しいログファイルを有効にするためには、その前にデータベースのデータをバックアップしなければなりません。このオプションを選択してデータベース設定ウィンドウを閉じると、バックアップが必要である旨の警告ダイアログが表示されます:




**OK**をクリックすると即座にバックアップが開始され、ログファイルが有効になります。**キャンセル**をクリックすると設定は保存されますが、ログファイルの作成は延期されます。この場合ログファイルは次回のバックアップ後に作成されます。

### 4Dのバックアップに関する留意点

4Dのバックアップメカニズムについてはデータベースのバックアップと復元を参照してください。

クライアント/サーバーモードでは、バックアップパラメターは4D Serverマシンでのみ設定できます。

#### Backup.XMLファイルを使用した設定の管理

すべてのデータベースバックアップおよび復旧の設定はXMLファイルに書き出されます。4Dはデータベース設定ダイアログボックスにオプションを表示するためと各バックアップが開始される際に、このファイル内のデータを使用します。

4DのXMLコマンドやXMLエディターを使用して設定内容を取得したり変更したりできます。これにより開発者は、特に4D Volume Desktopでビルドされたアプリケーションにおいて、バックアップ設定をプログラムで管理できます。

XML形式のバックアップおよび復旧設定ファイルはBackup.XMLという名前で、自動で以下の場所に作成されます:
*DatabaseFolder/Preferences/Backup/Backup.XML*
... *DatabaseFolder*はストラクチャーファイルが格納されているフォルダーです。サブフォルダー*Preferences/Backup/*はなければ自動で作成されます。バックアップの環境設定に使用されるXMLキーは*XML Keys-Backup*マニュアルで説明されています。

## バックアップ/バックアップ＆復旧ページ

このページのオプションを使用して、データベースのバックアップや自動復旧の一般的な設定を行うことができます。

### 一般設定

このエリアでは実際にバックアップが行われる際に実行される様々なメカニズムの設定を行います。

- **最新のバックアップのみ保存 X バックアップファイル**: このパラメーターを有効にすると、指定された数の最新バックアップファイルだけが保持され、古いバックアップファイルは削除されます。これによりディスクドライブの容量が不足することを避けることができます。
  この機能は以下のように動作します: バックアップ処理が完了したら、アーカイブが作成されたのと同じ場所、同じ名前のもっとも古いアーカイブを削除します。削除のタイミングを変更することもできます。
  例えば3世代のファイルを保持するよう設定している場合、最初の3回のバックアップでMyBase-0001、MyBase-0002、MyBase-0003が作成され、四回目のバックアップでMyBase-0004が作成されたのちにMyBase-0001が削除されます。デフォルトでこの設定は有効になっていて、4Dは3世代のバックアップを保持します。
  このメカニズムを無効にするには、チェックボックスの選択を外します。
  **注:** このパラメーターはデータベースバックアップセットとログファイルバックアップセット両方に関連します。
- **データファイルが更新された場合のみバックアップを行う**: このオプションが選択された場合、前回のバックアップ以降にデータが追加、変更、削除された場合のみ4Dは定期的なバックアップを開始します。そうでない場合定期的なバックアップはキャンセルされ、次回のスケジュールまで遅延されます。エラーは生成されませんが、バックアップジャーナルにはバックアップが遅延された旨記録されます。
  このオプションを使用すれば、原則的に参照目的で使用されているデータベースのバックアップに消費されるマシン時間を節約できます。ストラクチャーや添付ファイルに対して行われても、データファイルの更新としては扱われない旨注意してください。
  **注:** このパラメーターはデータベースおよびログファイル両方のバックアップに影響します。
- **最も古いバックアップファイルを削除**: このオプションは"最新のバックアップのみ保存 X バックアップファイル"が有効になっている場合のみ考慮されます。このオプションを使用して、最も古いバックアップファイルを削除するタイミングを設定します。選択肢はバックアップの開始前、あるいは完了後です。このオプションが有効に働くためには、バックアップファイルが名称変更されたり、移動されたりしていてはなりません。
- **バックアップ失敗時**: このオプションを使用してバックアップ失敗時の処理を設定できます。
  **注:** 定期的なバックアップが実行される予定時刻にデータベースが起動されていなかった場合、4Dはバックアップが失敗したものとして扱います。
  - **次回の予定された日付と時刻に再試行する**: このオプションは定期的な自動バックアップを設定されている場合にのみ意味があります。
  - **指定時間経過後に再試行**: このオプションが選択されていると、設定された待ち時間経過後にバックアップを再試行します。このメカニズムを使用すると、バックアップをブロックするような特定の状況に対応することが可能となります。秒、分、あるいは時間単位で待ち時間を設定できます。次のバックアップ試みにも失敗するとエラーが生成され、ステータスエリアに失敗状況が表示され、バックアップジャーナルに記録されます。
  - **操作をキャンセル X 試行後**: このパラメーターを使用して、バックアップ試行の失敗最大数を設定できます。
    この最大数に達してもバックアップが正しく実行できなかった場合、バックアップはキャンセルされエラー1401 ("バックアップ試行の最大数に達しました。自動バックアップは無効になります") が生成されます。この場合、データベースを再起動するか、手動バックアップが成功するまで自動バックアップは行われません。
    このパラメーターは、人による介入が必要となるような問題があり、バックアップ試行が自動的に繰り返されることにより全体的なパフォーマンスに影響するようなケースで使用できます。デフォルトでこのオプションは選択されていません。

### アーカイブ

このエリアではアーカイブ生成オプションを設定します。これらのオプションはメインのバックアップファイルとログバックアップファイルに適用されます。

- **セグメントサイズ (MB)**
  4Dではアーカイブをセグメントに分割できます。この振る舞いにより例えばバックアップファイルを複数の異なるディスク (DVDやZIP等) に格納できます。復元時、4Dは自動でセグメントを統合します。各セグメントにはMyDatabase[xxxx-yyyy].4BKといった名称がつけられます (xxxxはバックアップ番号、yyyyはセグメント番号)。例えばMyDatabaseデータベースのバックアップが3つのセグメントに分割されると、以下のような名前になります:
  MyDatabase[0006-0001].4BK、MyDatabase[0006-0002].4BK、MyDatabase[0006-0003].4BK
  **セグメントサイズ**はコンボボックスであり、各セグメントのサイズをMB単位で設定できます。メニューから定義済み値を選択するか、0~2048の値を入力できます。0を指定するとセグメント化はされません (**なし**を指定したのと同じ)。
- **圧縮率**
  デフォルトで4Dはバックアップファイルを圧縮し、ディスクスペースを節約します。しかし大量のデータがある場合、ファイルの圧縮処理はバックアップにかかる時間を長くします。
  **圧縮率**オプションを使用してファイルの圧縮モードを調整できます:
  - **なし**: ファイルの圧縮は行われません。バックアップは早く行われますが、ファイルサイズは大きくなります。
  - **速度** (デフォルト): このオプションはバックアップの速度とアーカイブサイズのバランスが考慮されたものです。
  - **圧縮率**: アーカイブに最大の圧縮率が適用されます。アーカイブファイルはディスク上で最小のサイズとなりますが、バックアップの速度は低下します。
- **インターレース率と冗長率**
  4Dは最適化 (インターレース) とセキュリティ (冗長) メカニズムに基づく特定のアルゴリズムを使用してアーカイブを生成します。これらのメカニズムを必要に応じて設定できます。これらのオプションのメニューには**低**、**中**、**高**、および**なし** (デフォルト) の選択肢があります。
  - **インターレース率**: インターレースとはデータを連続しない領域に書き込むことにより、セクター損傷の際のリスクを低減させるものです。率を上げることでリスクがより低減されますが、データの処理により多くのメモリが必要となります。
  - **冗長率**: 冗長は同じ情報を複数回繰り返すことで、ファイル中のデータを保護するものです。冗長率を高くするとよりファイルが保護されます。しかし書き込みは遅くなり、ファイルサイズも増大します。

### 自動復元

ここでは損傷したデータベースを開く際の自動メカニズムを設定します。

- **データベースが壊れていたら、最新のバックアップから復元する**: このオプションが選択されていると、異常が検知された場合、4Dは起動時に自動でデータベースの有効な最新のバックアップからのデータの復旧を開始します。ユーザーによる介入は必要ありませんが、処理はバックアップジャーナルに記録されます。
  **注:** 自動復元が行われた場合、データファイルのみが復元されます。添付ファイルやストラクチャーファイルを取得したい場合、手動の復元を行う必要があります。
- **データベースが完全でない場合、最新のログを統合する**: このオプションがチェックされると、プログラムはデータベースを開く際や復旧時に、自動でログファイルを統合します。
  - データベースを開く際に、データファイルに保存されていない処理がログファイル中に見つかった場合、4Dは自動でカレントログファイルを統合します。このようなケースはディスクに書き込まれていないデータがまだキャッシュ中に存在する状態で、電力の切断が起きた場合に発生します。
  - データベースの復元時、カレントログファイルや、バックアップファイルと同じ番号を持つログバックアップファイルが同じフォルダーに存在する場合、4Dはその内容を検証します。そしてデータファイルに書き込まれていない処理が見つかれば、自動で統合が行われます。

ユーザーにダイアログボックスが提示されることはありません。処理は完全に自動です。処理はバックアップジャーナルに記録されます。

### 4Dのバックアップに関する留意点

4Dのバックアップメカニズムについてはデータベースのバックアップと復元を参照してください。

クライアント/サーバーモードでは、バックアップパラメーターは4D Serverマシンでのみ設定できます。

#### Backup.XMLファイルを使用した設定の管理

すべてのデータベースバックアップおよび復旧の設定はXMLファイルに書き出されます。4Dはデータベース設定ダイアログボックスにオプションを表示するためと各バックアップが開始される際に、このファイル内のデータを使用します。

4DのXMLコマンドやXMLエディターを使用して設定内容を取得したり変更したりできます。これにより開発者は、特に4D Volume Desktopでビルドされたアプリケーションにおいて、バックアップ設定をプログラムで管理できます。

XML形式のバックアップおよび復旧設定ファイルはBackup.XMLという名前で、自動で以下の場所に作成されます:

*DatabaseFolder/Preferences/Backup/Backup.XML*

*... DatabaseFolder* はストラクチャーファイルが格納されているフォルダーです。サブフォルダー*Preferences/Backup/*はなければ自動で作成されます。バックアップの環境設定に使用されるXMLキーは*XML Keys-Backup*マニュアルで説明されています。

## クライアント-サーバー/ネットワークオプションページ

このページにはクライアント/サーバーモードでデータベースを使用する際に使用されるパラメーターが集められています。これらの設定はリモートモードでデータベースが使用されるときにのみ、使用されます。

パラメーターは2つのタブ (ネットワークオプションとIP設定) にグループ化されています。

### ネットワーク

#### 起動時にデータベースを公開する

このオプションを使用して、4D Serverデータベースが起動されると自動で公開データベースのリストに表示されるようにするかどうかを指定できます。

- このオプションが選択されていると (デフォルトオプション)、データベースは公開され、公開データベースの一覧に表示されます (リモート4D のTCP/IPページ)。
- このオプションがチェックされていないと、データベースは公開されず、公開データベースの一覧には表示されません。接続を行うには接続ダイアログボックスの**カスタム**ページに手作業でデータベースのアドレスを入力しなければなりません。

**備考:** この設定を変更したら、設定を有効にするためにサーバデータベースを再起動する必要があります。

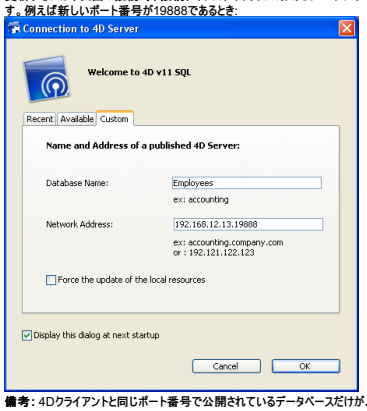
#### 公開名

このオプションでは、4D Serverがデータベースを公開する際に使用する名前を設定できます。この名前は例えば接続ダイアログボックスの利用可ページに表示されます (4D Serverデータベースへの接続参照)。
デフォルトで4D Serverはデータベースストラクチャファイル名を使用します。これを好きな名前に変更できます。

**備考:** このパラメタはカスタムクライアント-サーバアプリケーションでは使用されません。クライアントアプリケーションは接続ダイアログを経由せずにサーバに直接接続します。しかしエラーが発生すると、このダイアログが表示されます。このケースでは、サーバアプリケーションの公開名はコンパイルされたデータベースの名前です。

#### ポート番号

このオプションでは4D Serverのデータベース公開ポート番号を変更できます。この情報はデータベースのストラクチャ及びそれぞれのクライアントマシンに格納されます。4D Serverとリモートモードの4Dが使用するデフォルトのTCPポート番号は19813です。
TCPプロトコルを使用して、1台のマシン上で複数の4Dアプリケーションを同時に公開したい場合にこの変更が必要です。この場合、アプリケーションごとに異なるポート番号を割り当てなければなりません。
4D Server または 4Dからこの値を変更すると、データベースに接続しているすべての4Dマシンに変更が通知されます。接続していないクライアントを更新するには、次回の接続時、接続ダイアログボックスのカスタムページにサーバマシンのIPアドレスに続けてコロン、そして新しいポート番号を入力します。例えば新しいポート番号が19888であるとき:




**備考:** 4Dクライアントと同じポート番号で公開されているデータベースだけが、接続ダイアログの利用可ページに表示されます。

#### 4D Serverとポート番号

4D Serverは3つのTCPポートを使用して内部サーバとクライアントの通信を行います:

- **SQLサーバ**: デフォルトで19812 (環境設定の"SQL/設定"ページで変更可)。
- **アプリケーションサーバ**: デフォルトで19813 ("クライアント-サーバ/設定"ページで変更可)。
- **DB4Dサーバ**(データベースサーバ): デフォルトで19814。このポート番号を直接は変更できません。常にアプリケーションサーバのポート番号+1です。

4D クライアントが4D Serverに接続するとき、アプリケーションサーバのポート番号 (19813または接続ダイアログボックスのIPアドレス欄でコロンの後ろに指定された番号) を使用して接続します。サーバごとのポートを使用する他のサーバへの接続は自動です。再度ポート番号を指定する必要はありません。
ルータやファイアウォール越しに接続する場合、3つのポートを明示的に開く必要があります。

#### クライアント-サーバ接続タイムアウト

このサーモメータを使用して、4D Serverとクライアントマシン間で (一定時間活動が行われないときに接続を閉じる) タイムアウトを設定できます。
制限なしオプションはタイムアウトを設定しないことを意味します。このオプションが選択されると、クライアントのアクティビティコントロールは行われなくなります。
タイムアウトが選択されると、指定された時間リクエストを受け取らない時、サーバはそのクライアントとの接続を閉じます。

### クライアント-サーバ通信

#### Execute On Clientのために起動時にクライアントを登録

このオプションが選択されていると、データベースに接続するすべての4Dリモートマシン上で、リモートからメソッドを実行できます。このメカニズムについてはクライアントマシン上でのストアドプロシージャで説明しています。

#### クライアント-サーバ接続の暗号化

このオプションを使用して、サーバーマシンと4Dリモートマシン間通信のSSLモード (保護モード) を有効にできます。このオプションについてはクライアント/サーバ接続の暗号化で説明しています。

#### "Resources"フォルダをセッション中に更新

この設定は、セッション中にデータベースの**Resources**フォルダが更新されたとき、接続された4Dマシンの**Resources**フォルダのローカルインスタンスの更新モードをグローバルに設定するために使用されます(Resourcesフォルダはセッションが開かれるたびにリモートマシン上で自動的に同期されます)。
3つの選択肢があります:

- **しない**: ローカル**Resources**フォルダはセッション中に更新されません。サーバから送信される通知は無視されます。ローカルの**Resources**フォルダは**ローカルリソースを更新**コマンドを使用して手動で更新できます。
- **常に**: ローカル**Resources**フォルダの同期は、サーバから通知が送信されると、セッション中に自動で行われます。
- **その都度指定**: サーバから通知を受け取ると、クライアントマシン上でダイアログボックスが表示されます。ユーザはローカル**Resources**フォルダの同期を受け入れあるいは拒否できます。

**Resources**フォルダはデータベースインタフェースで使用されるカスタムファイルを格納するものです (翻訳ファイルやピクチャなど)。自動又は手動メカニズムを使用して、このフォルダの内容が更新されたときに各クライアントに通知できます。詳細はリソースフォルダの管理を参照してください。

#### ストラクチャを開くモード

このオプションは、クライアント側で通信をするときの介しモードをデータベースストラクチャに設定します。デフォルトは「読み／書き」モードです。「読み込みのみ」モードに設定すると、ストラクチャが更新されないようになります。

## クライアント-サーバー/IPオプションページ

### 許可-拒否設定表

この表を使用して、4DリモートマシンのIPアドレスに基づき、データベースへのアクセスコントロールルールを設定できます。このオプションを使用して、例えば戦略アプリケーションなどのセキュリティを高めることができます。

**Note:** この設定表でWeb接続はコントロールされません。

設定表の動作は以下のとおりです:

- "許可-拒否"列ではポップアップメニューを使用して適用するルールを選択します (許可または拒否)。ルールを追加するには、追加ボタンをクリックします。新しい行が表に追加されます。削除ボタンで選択した行を削除できます。
- "IPアドレス"列で、ルールに関連するIPアドレスを指定します。アドレスを指定するには、選択した行のセルをクリックし、以下の形式でアドレスを入力します: 123.45.67.89.
  * (アスタリスク) 文字をアドレスの末尾に使用して、範囲を指定することもできます。例えば192.168.* は192.168で始まるすべてのアドレスを示します。
- 表中のルールの順番は重要です。2つのルールが矛盾する設定の場合、より上に設定されているルールが適用されます。
  行の順番は列のヘッダをクリックしてソートを行ったり、ドラッグ&ドロップで移動したりして変更できます。
- セキュリティのため、ルールにより明示的に許可されたアドレスのみが接続を許可されます。言い換えれば、表に拒否ルールしか定義されていない場合、許可ルールに適合するアドレスがないため、すべてのアドレスからの接続が拒否されます。特定のアドレスからの接続のみを拒否したい場合 (そして他を許可したい場合)、許可 * ルールを表の最後に追加します。例えば:
  - 拒否 192.168.* (192.168で始まるアドレスを拒否)
  - 許可 * (他のアドレスはすべて許可)

  デフォルトでアクセス制限はありません。最初の行には * (すべてのアドレス) に対して許可が設定されています。

## Web/設定ページ

**Web**ページのタブを使用して、4Dに統合されたWebサーバーの様々な設定 (セキュリティ、開始オプション、接続、Webサービス等) にアクセスできます。4D Webサーバーの動作に関する詳細は4D *Language Reference*マニュアルのWebサーバを参照してください。Webサービスに関する詳細はWeb サービスの公開と使用を参照してください。

### 公開情報

#### 起動時にWebサーバを開始する

4Dアプリケーションの起動時にWebサーバを起動するか指定します。このオプションはWebサーバ設定と接続管理で説明しています。

#### TCPポート番号

デフォルトで、4Dは通常のWeb TCPポート番号である80番を使用してWebデータベースを公開します。他のWebアプリケーションによってこのポート番号が既に使用されている場合、4Dが使用するポート番号を変更する必要があります。Mac OS Xでは、TCPポート番号を変更することで、rootユーザを有効にしなくてもWebサーバを開始することができます。(Webサーバ設定と接続管理参照)
ポート番号を変更するには、**TCPポート入力**エリアに適切な値 (同じマシン上で動作する他のTCP/IPサービスが使用していないTCPポート番号)を指定します。

**Note:** 0を指定すると、4DはデフォルトのTCPポート番号80を使用します。

デフォルトでないTCPポート番号を使用して公開されたWebサーバに接続するには、Webブラウザに入力するアドレスに使用するポート番号を含めなければなりません。アドレスの後にコロンに続けてポート番号を指定します。例えばTCP番号8080を使用する場合、"123.4.567.89:8080"のように書きます。

**警告:** デフォルトのTCPポート番号 (標準モードで80、SSLモードで443) 以外を使用する場合、同時に使用する他のサービスのデフォルトポート番号を使用しないよう注意してください。例えば、WebサーバマシンでFTPプロトコルを使用する計画である場合、(あなたが何を行おうとしているか分かっていないのであれば) このデフォルトポートであるTCPポート20と21は使用してはいけません。標準のTCPポート番号割り当てを知るには、4D Internet CommandsのAppendix B - TCPポート番号の節を参照してください。256より下のポート番号はwell-knownサービスに予約されています。256から1024はUNIXプラットフォーム由来のサービスに予約されています。互換性のためにこれらの数値よりも上、例えば2000台や3000台などを指定します。

#### HTTPリクエストを受け付けるIPアドレスの指定

WebサーバがHTTPリクエストを受け付けるIPアドレスを指定できます。

デフォルトで、サーバはすべてのIPアドレスへのリクエストに応答します (**すべて**オプション)。
ドロップダウンリストにはマシンで利用可能なIPアドレスがリストされます。特定のアドレスを指定すると、サーバはこのIPアドレスへのリクエストにのみ応答します。

この機能は複数のIPアドレスが設定されたマシン上で動作する4D Webサーバのためのものです。これはしばしばインターネットホストプロバイダで使用されます。

このようなマルチホーミングシステムの実装はWebサーバマシン上での特定の設定を必要とします:

**Mac OSにセカンダリのIPアドレスをインストールする**

Mac OS上でマルチホーミングを設定するには:

1. **TCP/IP** コントロールパネルを開きます。
2. **設定** ポップアップメニューから**手動**オプションを選択します。
3. テキストファイルを"Secondary IP Addresses"という名称で作成し、SystemフォルダのPreferencesサブフォルダに保存します。
4. "Secondary IP Addresses"ファイルの各行にセカンダリIPアドレスと、オプションでサブネットマスクおよびセカンダリIPアドレス用のルータアドレスを記述します。

詳細はApple社のドキュメントを参照してください。

**WindowsにセカンダリのIPアドレスをインストールする**

Windows上でマルチホーミングを設定するには:

1. 以下のコマンド (またはWindows OSごとの同様の機能) を選択します:
   **スタート** メニュー > **コントロールパネル** > **ネットワークとインターネット接続** > **ネットワーク接続** > **ローカルエリア接続** (プロパティ) > **インターネットプロトコル (TCP/IP)** > **プロパティ** ボタン > **詳細...** ボタン。"詳細 TCP/IP 設定"ダイアログが表示されます。
2. "IPアドレス"エリアの**追加...** ボタンをクリックし、IPアドレスを追加します。
   最大5つまでIPアドレスを定義できます。これを行うにはシステム管理者に問い合わせる必要があります。詳細はWindowsのドキュメントを参照してください。

#### SSLを有効にする

Webサーバがセキュアな接続を受け入れるか受け入れないかを指定します。このオプションはSSLプロトコルの使用で説明します。

#### HTTPSポート番号

SSL (HTTPSプロトコル) を使用したセキュアなHTTP接続に対してWebサーバが使用するTCPポート番号を指定できます。デフォルトでHTTPSポート番号は443です。
この番号の変更を検討する主な理由は2つあります:

- セキュリティ: 攻撃者は標準のTCPポート (80 and 443) に集中して攻撃を仕掛けてくるかもしれません。
- Mac OS Xでは、特定のポート番号 (0 から 1023) を使用してWebサーバを起動するには、特別なアクセス権を必要とします。rootユーザだけがこれらのポートを使用するアプリケーションを起動できます。標準ユーザがWebサーバを起動するには、TCPポート番号の変更が一つのソリューションとなります (Webサーバ設定と接続管理 の節参照)。
  有効であれば何番でも渡すことができます (Mac OS X 上でのアクセス上の制限を避けるためには、1023より上の番号を指定します)。TCPポート番号に関する詳細は、先述の "TCPポート番号" を参照してください。

#### "4DSYNC" URLを許可する

データベース設定の"Web/設定"ページにあるこのオプションを使用して、/4DSYNC URLを含むリクエストのサポートを制限できます。このURLはHTTPを使用したデータの同期に使用されます (このメカニズムに関する詳細は4DSYNC/ URLを参照してください)。

このオプションを使用すれば/4DSYNC URLを含むリクエストの処理を有効/無効にできます:

- 選択されていない場合、/4DSYNC リクエストは標準のリクエストとみなされ、特別な処理を行いません。同期リクエストを行おうとすると"404 - resource unavailable"タイプのレスポンスが返されます。
- 選択されている場合、同期メカニズムが有効となります。/4DSYNC リクエストは特別なリクエストとして扱われ、4D HTTPサーバーが自動で解析とレスポンスを処理します。

デフォルトで:

- バージョン13以降で作成されたデータベースでは、このオプションは選択されていません。
- 互換性のため、以前のバージョンから変換されたデータベースではこのオプションが選択されています。HTTP同期機能を使用しない場合、このオプションの選択を解除してください。

このオプションのスコープはローカルアプリケーションであり、設定を変更した後はWebサーバーを再起動しなければなりません。

### パス

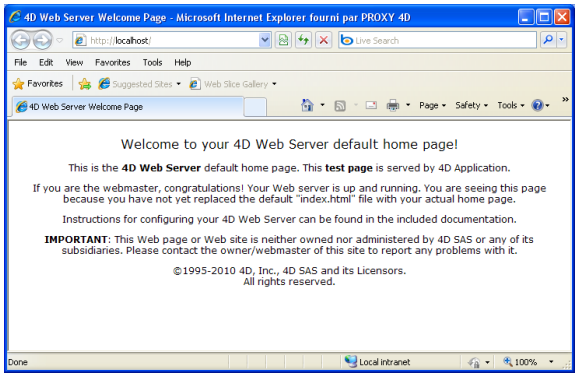
#### デフォルトHTMLルート

Webサイトファイルのデフォルトの位置を指定し、ディスク上でその場所より階層的に上のファイルにアクセスできない場所を指定します。このオプションは接続セキュリティの節で説明しています。

#### デフォルトホームページの設定

データベースに接続するすべてのブラウザー用にデフォルトホームページを指定できます。このページはスタティックでもセミダイナミックでも可能です。

デフォルトでWebサーバーが最初に起動されたとき、4Dは"index.html"という名前のホームページを作成し、HTMLルートフォルダーに置きます。この設定を変更しない場合、Webサーバーに接続するブラウザーには以下のようなページが表示されます:




デフォルトホームページを変更するには、ルートフォルダーの"index.html"ページを置き換えるか、使用するページのアクセス相対パスを"デフォルトホームページ"エリアに入力します。
アクセスパスはデフォルトHTMLルートからの相対パスで設定しなければなりません。
データベースのマルチプラットフォーム互換性を確保するために、4D Webサーバーは特別なアクセスパスの記法を使用します。シンタックスルールは以下のとおりです:

- フォルダーはスラッシュ ("/") で区切ります。
- アクセスパスはスラッシュ ("/") で終わっていてはなりません。
- フォルダー階層を1レベル上がるには、フォルダー名の前に "../" を記述します。
- アクセスパスはスラッシュ ("/") で始まっていてはいけません。

例えばデフォルトホームページを、デフォルトHTMLルートフォルダー内のWebサブフォルダーにある"MyHome.htm"にする場合、"Web/MyHome.htm"と入力します。

注: WEB SET HOME PAGEコマンドを使用して、Webプロセスごとにデフォルトホームページを設定できます。

デフォルトのカスタムホームページを指定しない場合、On Web Connectionデータベースメソッドが呼び出されます。プロシージャーでリクエストを処理するのは開発者の役割です。

## Web/オプション (I) ページ

#### スタティックページのキャッシュ

4D Webにはキャッシュがあり、スタティックページ, GIF, JPEG (<128 kb) そしてスタイルシート (.css files) などがリクエストされると、メモリにロードされます。
キャッシュの利用は、スタティックページの送信時にWebサーバのパフォーマンスを劇的に向上します。

キャッシュはすべてのWebプロセスで共有されます。環境設定でキャッシュのサイズを設定できます。デフォルトでスタティックページのキャッシュは無効になっています。有効にするには、**4D Webキャッシュを使用する**オプションをチェックします。

**ページキャッシュサイズ**エリアでキャッシュのサイズを変更できます。この設定はスタティックページのサイズや数、およびホストマシンで利用可能なリソースによります。

**注:** Webデータベースを利用する間、WEB GET STATISTICSコマンドを使用してキャッシュのパフォーマンスを検証できます。例えばキャッシュ利用率が100%に近い場合、キャッシュに割り当てたメモリ量を増やすことを考慮します。
/4DSTATS と /4DHTMLSTATS URLも、キャッシュの状態を知るために使用できます。Webサイトに関する情報を参照してください。

キャッシュが有効になると、4D Webサーバはブラウザからリクエストされたページをまずキャッシュから探します。ページが見つかると、即座にそれを送信します。見つからなければ、4Dはディスクからページをロードし、それをキャッシュに置きます。
キャッシュがいっぱいで、新しいページをキャッシュに置く必要がある場合、4Dは最も使われていないページの中から、もっとも古いページをアンロードします。

#### キャッシュのクリア

いつでもページやイメージをキャッシュからクリアできます (例えばスタティックページを更新し、キャッシュにそれをリロードさせたい場合)。
これを行うには、**キャッシュクリア** ボタンをクリックします。キャッシュは即座にクリアされます。

### Webプロセス

#### 非動作プロセスのタイムアウト

サーバー上で、活動していないWebプロセスを閉じるための最大タイムアウト時間を設定できます。

#### 最大同時Webプロセス

このオプションは、サーバー上で同時に開くことのできるすべてのWebプロセスの最大同時接続数を設定します(標準Webプロセス、またはプールされたプロセスなどすべて)。このパラメーターは、異常な数のリクエストによる4D Webサーバの飽和状態を避けるために使用します。

デフォルトでこの値は32000です。10から32000の値を設定できます。

最大Web同時接続数 (マイナス1) に達すると、4Dは新しいプロセスを作成せず、"Server unavailable" (ステータス HTTP 503 - Service Unavailable) を返信します。

#### 正しい値を決定する

理論上は、Webプロセスの最大数は以下の式で求められます: 利用可能なメモリ/Webプロセススタックサイズ
他のソリューションはランタイムエクスプローラでWebプロセスを監視することです。Webプロセスの現在数と、Webサーバが実行されてからの最大数が示されます。

#### 自動セッション管理

4D HTTPサーバーによるユーザーセッションの自動管理内部メカニズムを有効にしたり無効にしたりします。このメカニズムについてはWebセッション管理で説明しています。

4D v13以降に作成されたデータベースではデフォルトでこのメカニズムが有効になっています。他方互換性のため、v12以前のバージョンから変換されたデータベースでは無効になっています。この機能を利用するためには明示的に有効にしなければなりません。

このオプションが選択されていると、"一時的なコンテキストを再利用する"オプションも自動で選択され、ロックされます。

#### 一時的なコンテキストを再利用する (4Dリモートモード)

前のWebリクエストを処理するために作成されたWebプロセスを再利用することによって、4DリモートモードのWebサーバの動作を最適化できます。実際、4DリモートモードのWebサーバはそれぞれのWebリクエストを処理するために専用のWebプロセスを必要とします。必要であれば、このプロセスは4D Serverマシンに接続してデータやデータベースエンジンにアクセスします。そしてプロセス独自の変数やセレクションを使用してコンテキストを作成します。リクエストの処理が終了すると、このプロセスは廃棄されます。
**一時的なコンテキストを再利用する**オプションがチェックされていると、4Dは4Dリモートモード上で作成された固有のWebプロセスを保守し、その後のリクエストで再利用します。プロセスの作成処理が省略されるため、Webサーバのパフォーマンスが向上します。
他 方このオプションを使用する場合、不正な結果が返されることを避けるために、4Dメソッド内で使用される変数をシステマチックに初期化するようにしてください。同様に、以前のリクエストで使用されたカレントセレクションやカレントレコードをアンロードする必要があります。

### Web パスワード

#### BASICモードとDIGESTモード

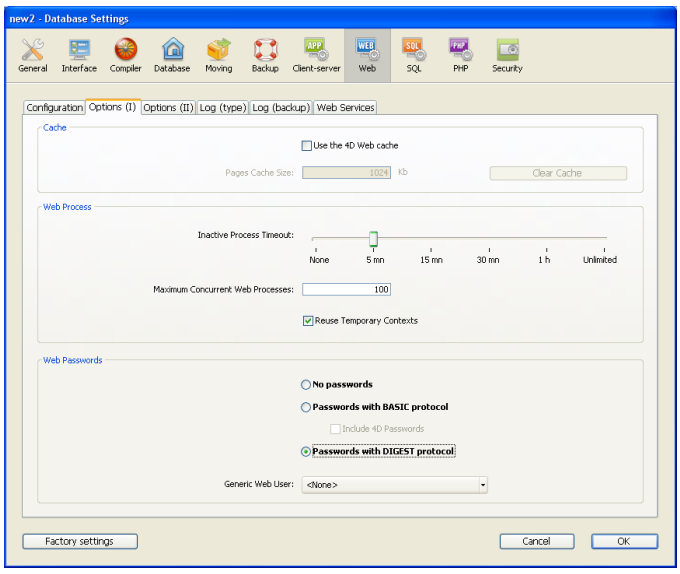
データベース設定で、Webサーバに適用するアクセスkんトロールシステムを設定できます。2つの認証モード、BASICモードとDIGESTモードが提供されています。認証モードはユーザ名とパスワードに関する情報の取得と処理方法に関連します。

- BASICモードでは、ユーザが入力したユーザ名とパスワードは暗号化されずにHTTPリクエストに含められて送信されます。この場合情報は第三者に盗聴・使用される可能性があるので、トータルなセキュリティは確保されません。
- DIGESTモードはより高いセキュリティレベルを提供します。認証情報は複合が困難な一方向ハッシュを使用して処理されます。

ユーザにとり、いずれの認証モードを使用するかは透過的です。

**Notes:**
互換性の理由から、バージョン11に変換された4Dデータベースでは、BASIC認証モードがデフォルトで使用 されます ("パスワードを使用" オプションが以前のバージョンでチェックされていれば)。Digestモードを使用するには明示的に指定しなければなりません。
Digest認証 はHTTP1.1の機能で、すべてのブラウザでサポートされているわけではありません。例えばバージョン5.0以降のMicrosoft Internet Explorerがこのモードを受け入れます。Digestモードが有効な時に、この機能をサポートしないブラウザがWebサーバにリクエストを送信すると、サーバはリクエストを拒否し、エラーメッセージをブラウザに返します。

データベース設定ダイアログボックスで、Webサーバに適用するアクセスコントロールシステムを定義できます。これを行うには**Web**テーマの**オプション (I)**ページを表示します:




"Webパスワード"エリアで、3つのオプションから選択できます:

- **パスワードなし**: Webサーバへの接続に認証を実行しない。この場合:
  - が 存在すれば、それが実行され、$1と$2に加え、ブラウザとサーバのIPアドレス ($3と$4) だけが提供されます。ユーザ名とパスワード ($5と$6) は空です。この場合、ブラウザのIPアドレスやリクエストされたサーバのIPアドレスを使用して、リクエストをフィルタできます。

- が存在しない場合、接続は自動で受け入れられます。
- **BASIC認証のパスワード**: BASICモードの標準認証です。ユーザがサーバに接続するとダイアログボックスがブラウザ上に表示され、ユーザ名とパスワードの入力を求められます。これら2つの値は他の接続パラメタ (IPアドレス、URI) とともにに送信され、開発者はそれを処理できます。

  **Note:** この場合、が存在しないと、接続は拒否されます。

  このモードを有効にすると、**4Dパスワードを含む**オプションが有効になり、カスタムパスワードシステムの代わりに、あるいは追加として使用できます。
- **DIGEST認証のパスワード**: DIGESTモードの認証。BASICモードのように、ユーザは接続時にユーザ名とパスワードを入力します。これら2つの値は暗号化されて、他の接続パラメタとともにに送信されます。この場合Validate Digest Web Password コマンドを使用してユーザを認証しなければなりません。

これらの設定を変更した場合は、Webサーバを再起動しなければなりません。
4Dリモートモードの Webサーバでは、すべての4Dリモートモードのマシンが同じユーザテーブルを共有することを覚えておいてください。ユーザ/パスワードの検証は4D Serverアプリケーションで行われます。

#### BASICモード: パスワードとOn Web Authentication データベースメソッドの組み合わせ

BASICモードを使用する場合、4D Webサーバーへの接続をフィルターするシステムは、2つのパラメーターの組み合わせに基づきます:

- データベース設定環ダイアログボックスのWebパスワードオプション
- On Web Authenticationデータベースメソッドの存在

組み合わせは以下のとおり:

"BASIC認証のパスワード"が選択され、"4Dパスワードを含む"が選択されていない場合:

- On Web Authenticationデータベースメソッドが存在すれば、それが実行され、すべての引数が渡されます。そしてユーザ名とパスワード、さらにブラウザーとWebサーバーのIPアドレスを使用して、より精密に接続をフィルターできます。
- On Web Authenticationデータベースメソッドが存在しない場合、接続は自動で拒否され、認証メソッドが存在しない旨のメッセージがブラウザーに送信されます。

**注:** ブラウザーから送信されたユーザー名が空の文字列で、かつOn Web Authenticationデータベースメソッドが存在しない場合、ブラウザーにパスワードダイアログボックスが送信されます。

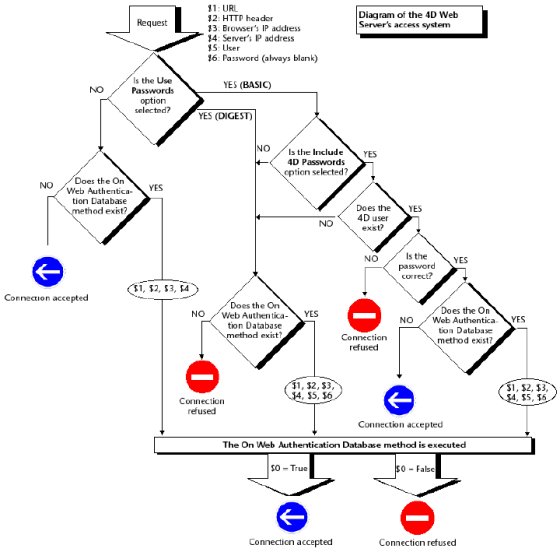
"BASIC認証のパスワード" と "4Dパスワードを含む" が選択されている場合:

- ブラウザーから送信されたユーザー名が4Dのユーザーテーブルに存在し、パスワードが正しい場合、接続は受け入れられます。パスワードが正しくなければ接続は拒否されます。
- ブラウザーから送信されたユーザー名が4Dに存在しない場合、2つの可能性があります:
  - On Web Authenticationデータベースメソッドが存在すれば、引数$1, $2, $3, $4, $5, $6に値が渡されます。ユーザー名とパスワード、さらにブラウザーとWebサーバーのIPアドレスを使用して、より精密に接続をフィルターできます。
  - On Web Authenticationデータベースメソッドが存在しない場合、接続は拒否されます。

#### DIGESTモード

BASICモードと異なり、DIGESTモードは標準の4Dパスワードシステムと互換がありません。4DパスワードをDigest Web認証に使用できません。このモードが選択されると"4Dパスワードを含む"オプションは選択不可となります。Webユーザーの認証は、(テーブル等を使用した) カスタマイズされた方法で管理しなければなりません。
DIGESTモードが有効の時、On Web Authenticationデータベースメソッドの$6 引数 (パスワード) は常に空の文字列が渡されます。実際このモードを使用するとき、この情報はネットワークからクリアテキストでは渡されません。この場合接続リクエストはWEB Validate digestコマンドを使用して検証しなければなりません。

4D Webサーバーのアクセスシステムの処理を以下に図示します:




### 一般Webユーザー

事前に4Dパスワードテーブルに登録したユーザーを、"一般Webユーザー"として定義できます。この場合サーバーに接続するブラウザーは、それぞれこの一般ユーザーに割り当てられたアクセス認証や制限を使用できます。これによりデータベースの異なる部分へのブラウザーによるアクセスを簡単にコントロールできます。

**注:** ブラウザーからメソッドやフォームなどアプリケーションの様々なパーツへのアクセスを制限することを可能にするこのオプションと、パスワードシステムとOn Web Authenticationデータベースメソッドで管理するWebサーバー接続コントロールシステムとを、混同しないでください。

一般Webユーザーを定義するには:

1. デザインモードでツールボックスのユーザーエディターを開き、最低1人のユーザーを作成する。ユーザーにパスワードを設定することもできます。
2. 他の4Dエディターを使用して、このユーザーに必要なアクセス認証あるいは制限を設定します。
3. データベース設定ダイアログで、**Web**テーマの**オプション (I)**ページを開く。
   "Webパスワード"エリアに**一般Webユーザー**ドロップダウンリストがあります。デフォルトで一般WebユーザーはDesignerであり、ブラウザーはデータベース全体にフルアクセスがあります。
4. ドロップダウンリストからユーザーを選択し、ダイアログを受け入れます。

データベースへの接続が認証されたすべてのWebブラウザーは、一般Webユーザーに割り当てられたアクセス認証と制限を使用します (BASICモードと"4Dパスワードを含む"オプションがチェックされ、接続したユーザーが4Dパスワードテーブルに存在しない場合を除く、以下参照)。

#### BASIC認証との相互作用

"BASIC認証のパスワード"は一般Webユーザーがどのように動作するかには影響を与えません。このオプションのステータスにかかわらず、データベースに接続が許可されたすべてのWebブラウザーに、"一般Webユーザー"に割り当てられたアクセス認証と制限が適用されます。

しかし"4Dパスワードを含む"オプションが選択されていると、2つの可能性が発生します:

- ユーザー名とパスワードが4Dパスワードテーブルに存在しない場合。この場合接続がOn Web Authenticationデータベースメソッドで受け入れられると、一般Webユーザーのアクセス権がブラウザに適用されます。
- ユーザー名とパスワードが4Dパスワードテーブルに存在する場合、"一般Webユーザー"は無視され、ユーザーは固有のアクセス権で接続します。

## Web/オプション(II)ページ

### テキスト変換

#### 拡張文字をそのまま送信

デフォルトで、4D Webサーバはダイナミックおよびスタティックページの拡張文字を、HTML標準に基づき送信前に変換します。このようなページはブラウザで評価されます。
拡張文字を変換せずそのまま送信するようWebサーバを設定できます。このオプションにより、特にShift_JIS文字コード利用時の日本語のシステムで速度が向上します。

この設定を行うには、**拡張文字をそのまま送信**オプションをチェックします。

#### スタンダードセット

**スタンダードセット**ドロップダウンリストを使用して、4D Webサーバが使用する文字セットを定義できます。デフォルトで文字セットはUTF-8です。4DのWebサーバはKeep-Alive接続を使用できます。Keep-Aliveオプションを使用して、一つの開かれたTCP接続でWebブラウザとサーバ間の一連のデータ交換を保守し、システムリソースを節約して、通信を最適化できます。

### Keep-Alive接続

**Keep-Alive接続を使用する**オプションは、WebサーバのKeep-Alive接続を有効および無効にします。このオプションはデフォルトで有効になっています。ほとんどの場合、通信が高速化されるため、この状態をお勧めします。WebブラウザがKeep-Alive接続をサポートしない場合、4D Webサーバは自動でHTTP/1.0にスイッチします。
4D WebサーバのKeep-Alive機能はすべてのTCP/IP接続 (HTTP, HTTPS) に関連します。しかしながらすべての4D Webプロセスで常にKeep-Alive接続が使用されるわけではないことに留意してください。あるケースでは、内部的な他の最適化機能が呼び出されることがあります。Keep-Alive接続は特にスタティックページで有効です。

Keep-Alive接続を設定する2つのオプションがあります:

- **接続毎のリクエスト数**: ひとつのKeep-Alive接続におけるリクエストとレスポンスの最大数を設定します。接続あたりのリクエスト数を制限することで、サーバのリクエスト過多を避けることができます (攻撃者が使用するテクニック)。
  4D Webサーバをホストするマシンのリソースに応じて、デフォルト値 (100) を増減できます。
- **タイムアウト**: この値を使用して、Webブラウザからのリクエストが行われない状態で、Webサーバが開かれた接続を保守する最大の待ち秒数を設定します。この秒数が経過すると、サーバは接続を閉じます。

接続が閉じられた後にWebブラウザがリクエストを送信すると、新しいTCP接続が作成されます。この動作はユーザからは見えません。

## Web/ログ (タイプ) ページ

### 接続ログファイル

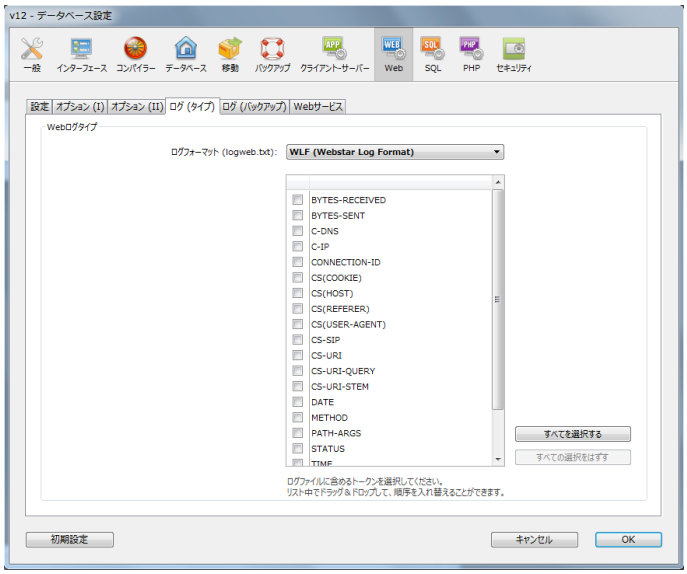
4Dはリクエストのログを提供します。

このファイルには、"logweb.txt" という名前が付けられ自動的に保存されます。

- 4Dのローカルモードと4D Serverでは、データベースストラクチャファイルと同階層のLogsフォルダに配置されます。
- 4Dのリモートモードでは、4Dクライアントデータベースフォルダ (キャッシュフォルダ) のLogsサブフォルダ内に配置されます。

### 有効化とフォーマット

ログファイルの有効化と内容の設定は、データベース設定のWeb/ログ (タイプ) ページで行います。




**注:** リクエストのログファイルの有効/無効はSET DATABASE PARAMETER (4D v12) または WEB SET OPTION (4D v13以降) コマンドを使用したプログラミングでも切り替えられます。

ログのフォーマットメニューでは、次のオプションを提供します。

- **No Log File**: このオプションが選択されると、4Dはリクエストのログファイルを作成しません。
- **CLF (Common Log Format)**: このオプションが選択されると、リクエストのログがCLFフォーマットで作成されます。CLFフォーマットでは、以下のように、それぞれのリクエストが行単位でファイル内に表示されます。
  host rfc931 user [DD/MMM/YYYY:HH:MM:SS] "request" state length

  各フィールドはスペースによって区切られ、それぞれの行はCR/LF シーケンス (character 13、character 10) で終ります。
  - host: クライアントのIPアドレス (例: 192.100.100.10)
  - rfc931: 4Dによって作成されない情報。常に - (マイナス記号) です。
  - user: 認証されているユーザ名、あるいは、- (マイナス記号) 。ユーザ名にスペースが含まれると、_ (下線) に置き換わります。
  - DD: 日、MMM: 月を表す3文字の略号 (Jan、Febなど)、YYYY: 年、HH: 時間、MM: 分、SS: 秒
    日付と時間はサーバのローカルタイム。
  - request: クライアントによって送られたリクエスト (例: GET /index.htm HTTP/1.0) 。
  - state: サーバからの返答。
  - length: 返されたデータ (HTTPヘッダー以外) のサイズまたは0

  **注:** パフォーマンス上の理由により、操作はメモリのバッファに1Kbのパケットで保存され、ディスクに書き込まれます。5秒間リクエストが発生しなくても、操作はディスクに書き込まれます。

  stateとして取り得る値は以下の通り。

  200: OK
  204: No contents
  302: Redirection
  304: Not modified
  400: Incorrect request
  401: Authentication required
  404: Not found
  500: Internal error

  CLFフォーマットはカスタマイズされません。
- **DLF (Combined Log Format)**: このオプションが選択されると、リクエストログがDLFフォーマットで作成されます。DLFフォーマットはCLFフォーマットと類似していて、全く同じ ストラクチャを使用します。各リクエストの最後に2つのHTTPフィールド、RefererとUser-agentを追加します。
  - Referer: リクエストされたドキュメントを指しているページのURLを含みます。
  - User-agent: オリジナルリクエストでクライアントのブラウザまたはソフトウェアの名 前とバージョンを含みます。

  DLFフォーマットはカスタマイズされません。
- **ELF (Extended Log Format)**: このオプションが選択されると、リクエストログがELFフォーマットで作成されます。ELFフォーマットはHTTPブラウザ界で広く普及しています。そし て、特別なニーズに応える洗練されたログを構築します。この理由により、ELFフォーマットはカスタマイズされます。レコードするフィールドやそのフィー ルドをファイルへ挿入する順番を選択することが可能です。
- **WLF (WebStar Log Format)**: このオプションが選択されると、リクエストログがWLFフォーマットで作成されます。WLFフォーマットは4D WebSTARサーバ用として特別に開発されました。いくつかの追加フィールドを持つので、EFLフォーマットと似ています。EFLフォーマットと同様、カスタマイズされます。

### フィールドの設定

ELF (Extended Log Format) フォーマットまたは WLF (WebStar Log Format) を選択すると、"Weg Log Token Selection" エリアで選択されたフォーマットに対して利用可能なフィールドが表示されます。ログに含む各フィールドを選択する必要があります。これを実行するには、矢 印ボタンを使用するか、"Selected Tokens" へ目 的のフィールドをドラッグ＆ドロップします。

**注**: 同じフィールドを2度選択することはできません。

各フォーマットで利用可能なフィールド (アルファベット順) とその内容を以下に示します。

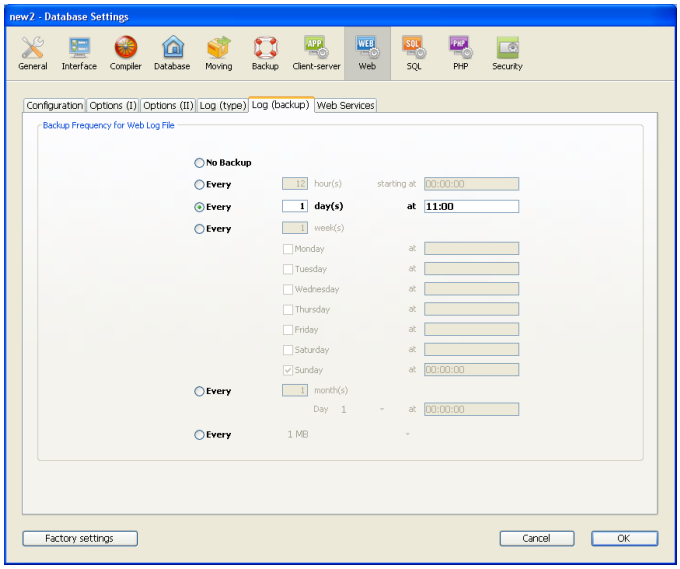
| フィールド | ELF | WLF | 値 |
|---|---|---|---|
| BYTES_RECEIVED | | X | サーバが受け取ったバイト数 |
| BYTES_SENT | X | X | サーバがクライアントに送ったバイト数 |
| C_DNS | X | X | DNSのIPアドレス (ELF: C_IP フィールドと同一のフィールド) |
| C_IP | X | X | クライアントのIPアドレス (例: 192.100.100.10) |
| CONNECTION_ID | | X | 接続ID番号 |
| CS(COOKIE) | X | X | HTTPリクエストに格納されているcookiesに関する情報 |
| CS(HOST) | X | X | HTTPリクエストのHostフィールド |
| CS(REFERER) | X | X | リクエストされたドキュメントを指すページのURL |
| CS(USER_AGENT) | X | X | ソフトウェアに関する情報とクライアントのオペレーティングシステム |
| CS_SIP | X | X | サーバのIPアドレス |
| CS_URI | X | X | リクエストが作成されるURI |
| CS_URI_QUERY | X | X | リクエストをクエリする引数 |
| CS_URI_STEM | X | X | クエリ引数のないリクエストのパート |
| DATE | X | X | DD: 日、MMM: 月を表す3文字の略号 (Jan、Febなど)、YYYY: 年 |
| METHOD | X | X | サーバへ送られるリクエスト用のHTTPメソッド |
| PATH_ARGS | | X | CGI引数: "$" の後に続く文字列 |
| STATUS | X | X | サーバの返答 |
| TIME | X | X | HH: 時間、MM: 分、SS: 秒 |
| TRANSFER_TIME | X | X | 返答を作成するためにサーバによってリクエストされた時間 |
| USER | X | X | 認証された場合はユーザ名。その他の場合は - (マイナス記号) 。<br>ユーザ名にスペースがある場合、_ (アンダーライン) に置き換えられる。 |
| URL | | X | クライアントがリクエストしたURL |

**注：** 日付と時間はGMTで表されます。

## Web/ログ (バックアップ) ページ

**バックアップパラメタ**

リクエストログの自動バックアップは、データベース設定のWeb/ログ (バックアップ) ページで設定します。




最初に、頻度 (日、週などの単位) またはファイルサイズの上限に対応するラジオボタンをクリックして選択します。必要に応じて、バックアップする正確な時間を指定します。

- **バックアップしない**: 設定されたバックアップ機能が無効になっています。
- **X 時間ごと**: 1時間単位でバックアップをプログラムする際、このオプションを使用します。1から24の間の値を入力します。
  - **starting at**: 最初のバックアップの開始時間を設定するために使用します。
- **X 日ごと**: 1日単位でバックアップをプログラムする際、このオプションを使用します。毎日バックアップを実行したい場合、1を入力します。このオプションをチェックすると、バックアップの開始時間を指定しなければなりません。
- **X 週ごと**: 1週間単位でバックアップをプログラムする際、このオプションを使用します。毎週バックアップを実行したい場合、1を入力します。このオプションをチェックすると、バックアップを開始する曜日と時間を指定しなければなりません。必要であれば、複数の曜日を選択することも可能です。例えば、水曜日と金曜日を 選択し、2つのバックアップを設定することができます。
- **X 月ごと**: 1ヶ月単位でバックアップをプログラムする際、このオプションを使用します。毎月バックアップを実行したい場合、1を入力します。このオプションをチェックすると、バックアップを開始する月においての日付と時間を指定しなければなりません。
- **サイズ指定**: カレントのリクエストログのファイルサイズに基づいてバックアップをプログラムする際、このオプションを使用します。ファイルが設定されたサイズに達した際、バックアップが自動的に起動します。サイズ制限は1、10、100または1000MBごとに設定可能です。

**Note:** バックアップの設定では、バックアップが開始する予定となっているときにWebサーバが起動していない場合、次回の起動において、4Dはバックアップが失敗したと見なし、データベース設定で示されている適切な設定を適用します。

## Web/Web サービスページ

このページのオプションを使用して4DアプリケーションのWebサービスを有効にし、(サーバーおよびクライアントの) 設定をすることができます。

4DにおけるWebサービスサポートについてはWeb サービスの公開と使用を参照してください。

**サーバー側設定**

このエリアでは4DをWebサービスサーバーとして使用するための設定を行います。

- **Webサービスリクエストを許可する**: このオプションを使用してWebサービスクライアントからのリクエストを受け付けるかどうかを設定します。このオプションが選択されていない場合、4DはSOAPリクエストを拒否し、またWSDLを生成しません。このオプションを選択すると、4Dはメソッドプロパティに基づきWSDLファイルを生成します。
- **Webサービス名**: このエリアではWebサービスの"包括的な名称"を変更できます。この名前はSOAPサーバーレベルでサービスを識別するために使用されます。デフォルトで4Dは"A_WebService"を使用します。
- **Webサービス名前空間**: このエリアは4Dが公開するWebサービスの名前空間を設定するために使用します。
  インターネットに公開される各Webサービスはユニークでなければなりません。Webサービス名をユニークにするためにXML名前空間が使用されます。名前空間は任意の文字列で、XMLタグをユニークに識別するために使用されます。典型的には、名前空間は会社のURLで始まります (http://mycompany.com/mynamespace)。指定された文字列がユニークである限り、指定したURLに何か付け加える必要はありません。
  デフォルトで4Dは以下の名前空間を使用します: http://www.4d.com/namespace/default.
  **注:** タグ名に関するXML標準に準拠するために、使用する文字には空白が含まれていてはなりません。また数字から始まっていてはなりません。また非互換性を避けるためにアクセント文字や日本語などの拡張文字を使用しないことを推奨します。

**クライアント側設定**

このエリアではWebサービスクライアントとして4Dを使用するための設定を行います。

- **ウィザードメソッドプリフィクス**: このエリアでは、Webサービスウィザードを使用して4Dが自動で生成するプロクシメソッドの接頭辞を設定します。
  プロクシプロジェクトメソッドは4DアプリケーションとWebサービスサーバーとのリンクを形成します。
  デフォルトで"proxy_"が使用されます。

## SQLページ

このページでは4Dに統合されたSQLサーバーのデフォルト公開パラメーターやアクセス権、および4D SQLエンジンの動作に関する設定を行います。
4DにおけるSQLの詳細についてはSQLリファレンスマニュアルのMissingRefを参照してください。

**SQLサーバの公開設定**

4Dに統合されたSQLサーバの公開パラメタを変更できます。このパラメタはデータベース設定の**SQL/設定**ページにあります:

- **開始時にSQLサーバを起動**オプションを使用して、アプリケーションの開始時にSQLサーバを起動できます。
- **TCPポート**: デフォルトで4D SQLサーバはTCPポート19812に応答します。このポートが他のサービスで既に使用されていたり、あなたの接続パラメーターが他の設定を必要とする場合 は、4D SQLサーバが使用するポート番号を変更できます。
  **Note**: 0を指定すると、4DはデフォルトのTCPポート19812を使用します。
- **IPアドレス**: SQLサーバがSQLクエリを処理するために使用する、マシンのIPアドレスを限定することができます。デフォルトでは**すべて**が選択されて いて、サーバはすべてのIPアドレスへのクエリに応答します。
  "IPアドレス"ドロップダウンリストには、マシンに設定されているすべてのIPア ドレスがリストされます。特定のアドレスを選択すると、サーバはこのIPアドレスに送信されたクエリのみに応答します。
  これは複数のTCP/IP アドレスを持つマシン上で動作する4Dアプリケーションで有効な設定です。
  **Note**: クライアント側では、アプリケーションが接続するSQLサーバのIPアドレスやポート番号を、ODBCデータソース定義で正しく設定しなければなりません。
- **SSLを有効にする**: このオプションは、SQL接続を処理する際にSQLサーバがSSLプロトコルを使用するかしないかを指定します。このオプションをチェックした時は、**key.pem**と**cert.pem**ファ イルを以下の場所へコピーしてください:
  MyDatabase/Preferences/SQL ("MyDatabase"はデータベースフォルダ/パッケージ)。
  この場合SQL LOGINコマンドで接続を開く際、SQLサーバーのIPアドレスの後ろに":ssl"キーワードを追加しなければならない点に留意してください。
- **Flash Playerリクエストを許可する**: このオプションで、4D SQLサーバによるFlash Playerリクエストをサポートするメカニズムを有効にするかどうかを指定できます。このメカニズムは、データベースの環境設定フォルダ (Preferences/SQL/Flash /) に "socketpolicy.xml" という名前のファイルが存在するかどうかに基づきます。このファイルはFlash Playerがクロスドメインの接続やFlex (Web 2.0) アプリケーションのソケットによる接続を行う場合に必要とされます。
  以前の4Dバージョンでは、このファイルを手 作業で追加しなければなりません でした。これからは、**Flash Playerリクエストを許可する**オプションをチェックすることにより有効化されます。このオプションを チェックすると、Flash Playerリクエストが受け入れられ、必要に応じて、データベース用の汎用"socketpolicy.xml" ファイルが作成されます。このオプションの選択を解除した場合、 "socketpolicy.xml" ファイルは無効となります（名前変更）。後にSQLサーバが受け取るすべてのFlash Playerクエリは拒否されます。
  データベースを開い た時、このオプションがチェックされているか否かは、データベースの環境設定フォルダに有効な"socketpolicy.xml" ファイルが存在するかどうかに基づきます。
  **Note**: 4DのSQL SET OPTIONコマンドで、外部リクエストを処理する際にSQL サーバが使用するエンコーディングを設定できます。

**デフォルトスキーマに対するSQLアクセスコントロール**

セキュリティの理由で、SQLサーバに送信された外部クエリが4Dデータベースで行うことのできるアクションを制限できます。
2つのレベル でコントロールが可能です:

- 許可するアクションのタイプのレベル
- クエリを実行するユーザのレベル
- これらはデー タベース設定の**SQL/設定**ページで設定できます

**備考:** On SQL Authenticationデータベースメソッドを使い、4D内部SQLエンジンに対する全ての外部アクセスを制御することもできます。

このダイアログボックスに設定されているパラメタはデフォルトスキーマに適用されます。データベースへの外部接続制御 は、SQLスキーマ (4Dと4D SQLエンジン統合の原則参照） のコンセプトに基づきます。カスタムスキーマを作成しない場合、すべてのデータベーステーブルはデフォルトスキーマに含まれます。特定のアクセス権で他の スキーマを作成し、それらをテーブルに割り当てる場合、カスタムスキーマに含まれていないテーブルのみが、デフォルトスキーマに含まれます。

SQLサーバによるデフォルトスキーマに対し、3タイプのアクセス設定ができます:

- "読み込みのみ (データ)": データベーステーブルのすべてのデータに対する無制限の読み込みアクセス。レコードの追加、更新、削除のほか、データベーストラクチャに対する変更はで きません。
- "読み書き (データ)": データベーステーブルのすべてのデータへの読み書き (追加、更新、削除) アクセスが可能です。データベーストラクチャの更新は許可されません。
- "フルアクセス (データとデザイン)": データベーステーブルのすべてのデータへの読み書き (追加、更新、削除) およびデータベーストラクチャの更新 (テーブル、フィールド、リレーションなど) が許可されます。

それぞれのアクセスタイプごとにユーザのセットを割り当てることができます。3つのオプションが選択可能です:

- <**Nobody**>: このオプションを選択すると、指定されたタイプのアクセスは、クエリの発行元にかかわらず拒否されます。このパラメタは4Dパスワードアクセス管理システ ムが有効でない場合でも使用できます。
- <**Everybody**>: このオプションを選択すると、指定されたタイプのアクセスは、制限なしにすべてのクエリに対して許可されます。
- ユーザーグループ: このオプションは、ユーザのグループを作成し、特定のタイプへのアクセスを行うことができるユーザを指定するために使用します。このオプションを使用する ためには4Dパスワードシステムが有効になっていなければなりません。クエリ元のユーザは、SQLサーバへの接続時、ユーザ名とパスワードを提示します。

**警告**: このメカニズムは4Dパスワードシステムに依存します。SQLサーバに対するアクセスコントロールを動作させるためには、(Designerにパスワードを設定して、) 4Dパスワードシステムが有効になっていなければなりません。

**Note**: 4Dプロジェクトメソッドごとに追加のセキュリティオプションを設定できます。詳細は4Dと4D SQLエンジン統合の原則の"SQL利用可オプション"を参照してください。

**SQLエンジンオプション**

- **自動コミットトランザクション**: このオプションを使用してSQLエンジンの自動コミットメカニズムを有効にできます。自動コミットモードの目的は、データの参照整合性を保つことにあります。このオプションがチェックされていると、トランザクションの中で実行されていないすべてのSELECT、INSERT、UPDATE、そしてDELETE (SIUD) クエリは、自動でアドホックなトランザクションに含められます。これによりクエリが完全に実行されるか、エラーが発生した場合にはキャンセルされることが 保障されます。
- すでにトランザクションの中にある (参照整合性がカスタムに管理されている) クエリは、このオプションの影響を受けません。
  この オプションにチェックがされていないと、(**SELECT... FOR UPDATE** クエリを除き) 自動トランザクションは生成されません (SELECTコマンド参照)。デフォルトでこ のオプションはチェックされていません。
  SET DATABASE PARAMETERコ マンドを使用して、プログラムでこのオプションを管理することもできます。
  **Note**: 4D SQLエンジンによりクエリされるローカルデータベースのみがこのパラメタの影響を受けます。外部データベースの場合、自動コミットメカニズムはリモート のSQLエンジンが処理します。
- **大文字小文字を区別した文字列比較**: このオプションを使用してSQLクエリ時の文字列比較を変更することができます。このオプションはデフォルトでチェックされていて、SQLエンジンは文字 列比較 (ソートやクエリ) の際に大文字小文字を区別します。 例えば"ABC"="ABC" ですが "ABC" # "Abc" となります。
  SQL エンジンと4Dエンジンの動作を揃えたいなど特定のケースでは、文字列比較で大文字小文字を区別したくない ("ABC"="Abc") 場合があります。そのようにするには、このオプションの選択を外します。
  SET DATABASE PARAMETERコマンドを使用して、プログラムでこのオ プションを管理することもできます。

## PHPページ

4Dから直接PHPスクリプトを実行できます。このページではPHP実行に関する設定を行います。PHPの実行については4Dランゲージリファレンスの4DでPHPスクリプトを実行する参照してください。

**インタープリター**

- **IPアドレス と ポート番号**
  4Dはデフォルトで使用されるFastCGI PHPインタープリターを提供しています。内部的なアーキテクチャーに関連する理由により、実行リクエストは特定のHTTPアドレスのPHPインタープリターに送信されます。4Dはデフォルトでアドレス127.0.0.1そしてポート8002を使用します。他のサービスですでにこのアドレスやポートが使用されている場合、あるいは同じマシン上で複数のインタープリターが動作する場合、設定を変更する必要があります。これを行うには**IPアドレス**と**ポート番号**パラメーターを変更します。
  HTTPアドレスは4Dと同じマシンでなければならない点に注意してください。

- **外部インタープリター**
  外部のPHPインタープリターを使用する場合、それはFastCGIでコンパイルされ、4Dと同じマシン上になければなりません (4DでPHPスクリプトを実行するの"異なるPHPインタープリターとphp.iniを使用する"参照)。
  このオプションを選択すると4Dは内部PHPインタープリターに接続しなくなります。この設定を行う場合、外部インタープリターの制御を開発者が行わなくてはならない点に留意してください。

**4D Server:** この設定は、4D Serverとリモートの4Dで共有されます。サーバーマシンで外部インタープリターを利用する時は、クライアントマシンで内部インタープリターを利用することはできません。もしサーバーがポート9002で外部インタープリターを利用する場合には、クライアントマシンでも同じポート番号でインタプリターを利用しなければなりません。

**オプション**

このエリアのオプションは4Dの内部PHPインタープリターの自動管理に関するものであり、**外部インタープリターオプション**が選択されているときは無効になります。

- **プロセス数**: 4DのPHPインタープリターは"子プロセス"と呼ばれる一連のシステム実行プロセスを起動します。最適化のためデフォルトで同時に5つの子プロセスを保持し実行できます。必要に応じて子プロセスの数を変更できます。例えばPHPインタープリターを頻繁に呼び出す場合、数を増やしたいと思うかもしれません。この点に関する詳細は4DでPHPスクリプトを実行するのアーキテクチャーを参照してください。
  **注:** Mac OSではすべての子プロセスが同じポートを共有します。Windowsでは子プロセスごとにポート番号が異なります。最初に使用されるポートはPHPインタープリター用に設定されたポートです。そして次の子プロセスはインクリメントされた番号を使用します。例えばデフォルトポート番号が8002で5つの子プロセスを起動すると、8002から8006が使用されます。

- **インタープリターを再起動 Xリクエスト後**: 4DのPHPインタープリターが受け付けるリクエストの最大数を設定します。実行リクエストがこの数に達すると、インタープリターが再起動されます。このパラメーターに関する詳細はFastCGI-PHPのドキュメントを参照してください。
  **注:** このダイアログボックスではすべての接続されたマシンおよびすべてのセッションのデフォルト値を設定します。各マシンおよび各セッションで異なる設定を適用するためにSET DATABASE PARAMETERとGet database parameterコマンドを使用できます。SET DATABASE PARAMETERコマンドで変更された値はカレントセッションで使用されます。

## セキュリティページ

このページではデータへのアクセスやデータベースの保護に関する設定を行います。

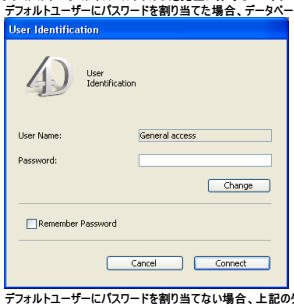
**データアクセス権**

- **デザインアクセス権**: データベースのデザインモードへの特定のグループに与えるために使用します。
  以下の点に留意してください:
  - デザインモードへのアクセスグループを設定すると、データ読み込みダイアログの**テーブルを作成**オプションが無効となります。このダイアログボックスに関する詳細はファイルからデータを読み込むを参照してください。
  - DesignerとAdministratorは常にデザインモードへアクセスできます。設定されるアクセスグループのメンバーに含められる必要はありません。

  ユーザーおよびグループに関する詳細はユーザー＆グループを参照してください。

- **フォーミュラエディターでコマンドやプロジェクトメソッドをフィルター**:
  セキュリティのため4Dはデフォルトで、アプリケーションモードでのでコマンド、関数、プロジェクトメソッドへのアクセスを制限しています。SET ALLOWED METHODSコマンドを使用して明示的に許可された4D関数やプロジェクトメソッドのみを使用することができます。以下のオプションを使用して部分的あるいは全体的にこのフィルタリングを無効にできます。
  - **すべて有効** (デフォルトオプション): DesignerとAdministratorを含むすべてのユーザーに対し、アプリケーションモードにおいてコマンドや関数、プロジェクトメソッドへのアクセスを制限します。
  - **DesignerとAdministratorのとき無効にする**: このオプションはDesignerとAdministratorのみに、フォーミュラエディターでの4Dコマンドやメソッドへの完全なアクセスを与えます。他のユーザには制限をかけつつ、データベースの管理者に無制限のアクセスを与えたい場合に使用できます。開発中にはこのモードを使用してすべてのフォーミュラやレポート等をテストできます。運用中には一時的にコマンドやメソッドへのアクセスを与えるなどのために使用できます。これを行うためには、コマンドへのフルアクセスが必要なダイアログを呼び出したり印刷処理を開始したりする前に (CHANGE CURRENT USERコマンドを使用して) ユーザーを切り替えます。そしてその処理が終了したら元のユーザーに戻します。
    **注:** 前のオプションを使用してフルアクセスが有効にされると、このオプションは効果を失います。
  - **すべて無効**: このオプションはフォーミュラエディターでのユーザーアクションの制御を無効にします。このオプションが選択されるとユーザーはすべての4Dおよびプラグインコマンド、さらにはプロジェクトメソッドを使用できます (非表示のものを除く)。
    **注:** このオプションはSET ALLOWED METHODSコマンドより優先されます。このオプションが選択されるとコマンドの効果はなくなります。

- **デフォルトユーザー**: デフォルトユーザーが設定されると、データベースを開く、あるいはデータベースにログインするすべてのユーザーは、このデフォルトユーザーに定義された同じアクセス権と制限を持つことになります。さらにデフォルトユーザーにパスワードを割り当てない場合、パスワードダイアログボックスは表示されず、データベースが直接開かれます。
  このオプションはデータコントロールシステムを完全に保守しつつ、データベースへのアクセスを簡易化することができます。
  - デフォルトユーザーにパスワードを割り当てた場合、データベースが開かれるときにダイアログが表示され、パスワードの入力を求められます:

  - デフォルトユーザーにパスワードを割り当てない場合、上記のダイアログは表示されません。

  **注:**
  - このモードを有効にするためにはパスワードシステムを有効にする必要があります。
  - デフォルトユーザーモードが有効になっているときでも、強制的にユーザー認証ダイアログを表示させることができます。これは例えばAdministratorやDesignerとしてログインするために必要となります。これを行うにはデータベースを開いたり接続したりする際に**Shift**キーを押したままにします。

- **パスワードダイアログにユーザーリストを表示する**: このオプションが選択されていると、ユーザー認証ダイアログにユーザーリストが表示され、ユーザーはその中から名前を選択し、パスワードを入力することになります。選択されていない場合、名前とパスワード両方を入力します。パスワードダイアログボックスの2つのバージョンに関する詳細はアクセスシステムの概要の"アクセスシステムの概要"を参照してください。
  - **ユーザーリストをABC順で表示する** (上記オプションが選択されているときのみ有効です): このオプションが選択されていると、ユーザー認証ダイアログボックスのユーザーリストは名前順に表示されます。

- **ユーザーは自分のパスワードを変更可能**: このオプションが選択されていると、ユーザー認証ダイアログに**変更**ボタンが表示されます。このボタンを使用するとパスワードを変更するためのダイアログボックスが表示されます (このダイアログに関する詳細はパスワードアクセスシステムの保守の"ユーザーによるパスワードの変更"を参照してください。)。必要であれば**変更**ボタンを非表示にし、パスワードの変更を禁止することができます。そうするためにはこのオプションの選択を外します。

**Options**

- **外部ファイルのユーザー設定を有効にする**: 外部ファイル化したユーザー設定を使用するためにはこのオプションを選択します。このオプションが選択されると、設定を行うダイアログが2つになります: **ストラクチャー設定**と**ユーザー設定**。詳細はユーザー設定を使用するを参照してください。

- **コンポーネントの"On Host Database Event" メソッドを実行**: On Host Database Event データベースメソッド は4D コンポーネントの初期化とバックアップフェーズを容易にします。セキュリティ上の理由から、このメソッドの実行はそれぞれのホストデータベースにおいて明示的に認証されなければなりません。そのためにはこのオプションをチェックします。デフォルトでは、チェックされていません。
  このオプションがチェックされていると:
  - 4D コンポーネントがロードされます。
  - コンポーネントのそれぞれの On Host Database Event データベースメソッド がホストデータベースによって呼び出されます。
  - メソッドのコードが実行されます。

  このオプションがチェックされていないと:
  - 4D コンポーネントはロードはされるものの、スタートアップとシャットダウンは行われません。
  - コンポーネントのデベロッパは、これらのフェーズの間ホストデータベースによって呼び出される必要のあるコンポーネントについて、コンポーネントメソッドを公開しなければなりません(スタートアップとシャットダウン)。
  - ホストデータベースのデベロッパは、コンポーネントの適切なメソッドを適切なタイミングで呼び出さなければなりません(適切なタイミングとコンポーネントは、デベロッパが提供します。)。

## 互換性ページ

互換性ページには、以前の4Dバージョンとの互換性を管理するためのパラメーターがまとめられています。ここに表示されるオプションの数は、元の データベースが作成されたバージョン (2004.x, v11, v12等) や、このデータベースで行われた設定の変更により異なります。

**注:** このページは現在のバージョンで作成されたデータベース (変換されていないデータベース) には表示されません。

- **ダイアログボックスでフィールドを入力可にする**: 以前のバージョンではDIALOG コマンド等で表示されたダイアログボックスでフィールドに値を入力することができませんでした。この制限は4D 2004で取り除かれています。データを表示するだけの目的でダイアログにフィールドを表示している場合、以前の動作を保持することができます。バージョン2004に変換されたデータベースではこのオプションが選択され、v2004で作成されたデータベースでは選択されていません。

- **ラジオボタンを名前でグループ化する**: 以前のバージョンではラジオボタンの排他制御はボタンに割り当てる変数名の先頭バイトで判定されていました (例えば*m_button1, m_button2, m_button3*等)。4D 2004以降はフォームエディター上でオブジェクトをグループ化することで排他制御が行われるようになっています。この点についてはラジオボタンとピクチャーラジオボタン を参照してください。
この新しいモードはラジオボタン、3Dラジオボタン、ピクチャーラジオボタンで有効です。互換性を保 ために、変換されたデータベースでは以前のモードが使用されています。しかしこのオプションの選択を解除すれば新しいモードを使用できます。v2004 で作成されたデータベースでは新しいモードが使用されます。
- **PRINT SELECTION中、レコード毎にフォームをリロード**: 以前のバージョンの4DではPRINT SELECTIONコマンドを使用した印刷中に使用されるフォームは、各レコード毎にリロードされていました。これによりOn printing detailフォームイベントで開発者が言語を使用して変更したかもしれないオブジェクトの設定がすべて自動的に再初期化されていました。
パフォーマンスを最適化するためにこのメカニズムは4D 2004で取り除かれました。今後はフォームメソッドを使用して4D開発者が初期化を行わなければなりません。この動作はOn display detailフォームイベントを使用するリストフォームと同じです。しかしながらこのオプションを使用して以前の動作を保持することができます。v2004で作成されたデータベースは新しいモードを使用します。
- **ブラケットの代わりに4DVARコメントを使用する**: このオプションで、4Dタグを使用した4D式の挿入方法を指定します。このオプションが選択されている場合 (デフォルト)、標準のHTML記法 (<!--4DVAR MAVAR-->) を使用します。
オプションが選択されていない場合、ブラケット記法 ([MAVAR]) を使用します。この記法は以前のバージョンの4D Webサーバーで使用されていたプロプリエタリな方法であり、推奨されません。
- **新しいコンテキスト参照モードを使用しない**: このオプションが選択されていない場合、4D WebサーバーはHTMLのベースURLにコンテキスト番号を挿入します。
以前のモードでは、4D Webサーバーはブラウザーに送信する各項目にコンテキスト番号を送信しており、結果処理が遅くなっていました。しかしながら互換性のためこのオプション が選択されているかもしれません。このオプションを変更した後は設定を有効にするためにデータベースを再起動しなければなりません。
- **未知のURLから"/"を取り除く**: 以前の4Dではディスク上に存在しないファイルがURLとしてリクエストされた場合、On Web AuthenticationやOn Web Connectionデータベースメソッドの$1引数に先頭の"/"が取り除かれたURLが渡されていました。この動作は4D 2004で変更されました。しかし以前の動作に基づいた実装を行っている場合にはこのオプションを選択します。
- **外部からのドラッグ＆ドロップを拒否**: v11以降、ピクチャーなどのファイルや選択されたテキストオブジェクトなどを4Dにドラッグ＆ドロップできるようになりました。変換されたデータベース ではこの動作を想定したメソッドが書かれていないために期待した動作とならないかもしれません。このオプションを選択すると外部オブジェクトを4Dフォー ムにドロップできなくなります。ただしこの場合でも**自動ドロップ**オプションを使用すると外部オブジェクトの挿入が可能である点に留意してください。アプリケーションはドロップされたテキストやピクチャーを解釈します (ドラッグ＆ドロップ参照)。
- **QUERY BY FORMULAをサーバー上で実行とORDER BY FORMULAをサーバー上で実行**: 4D v11より最適化の目的で、フォーミュラによるクエリや並び替えコマンドがサーバー上で実行されるようになりました。そして結果だけがクライアントマシンに返されます。この動作は以下のコマンドで有効です: QUERY BY FORMULA、QUERY SELECTION BY FORMULA、ORDER BY FORMULA。変数が直接フォーミュラ内で使用されている場合、クライアントマシン上の変数値を使用してフォーミュラが呼び出されます。例えば
```
QUERY BY FORMULA([aTable];[aTable]aField=theVariable)
```
このコードがサーバー上で実行された場合でも、*myvariable*変数値はクライアントマシン上のものが使用されます。他方この原則はフォーミュラにメソッドが使用され、そのメソッド内で変数が参照されている場合には当てはまりません。この場合サーバー上で変数が解釈されます。
変換されたデータベースではこの点が考慮されていない可能性があるため、デフォルトでこれらのコマンドはクライアントマシン上でフォーミュラを実行します。新しいモードを使用したい場合はこれらのオプションを明示的に選択します。
**注:** このオプションはSET DATABASE PARAMETERコマンドで設定することもできます。
- **QUERY BY FORMULAでSQL JOINを使用**: 4D v11よりQUERY BY FORMULAやQUERY SELECTION BY FORMULAコマンドはSQLの結合モデルに基づくJOINを実行するようになりました。これによりストラクチャーエディターで自動リレーションが設定されていなくても[Table_A ]field_X=[Table_B ]field_Yのようなフォーミュラを使用できるようになりました。
既存のデータベースでこの動作が考慮されていない場合、予期しない動作となることがあるため、変換されたデータベースではこの機能がデフォルトで無効にされています。データベースコードを見直した後、このモードを有効にすることを推奨します。
**注:**
  - "SQL JOIN"モードが有効な場合でも、以下のケースではQUERY BY FORMULAやQUERY SELECTION BY FORMULAコマンドはストラクチャーエディターで設定された自動リレーションを使用します:
    - フォーミュラを{field ;comparator ;value}形式に分解できない場合
    - 同じテーブルの2つのフィールドが比較されている場合
  - SET DATABASE PARAMETERコマンドを使用してプロセス毎にこのオプションを設定できます。
- **トランザクションのネストを許可する**: マルチレベルトランザクションのサポートを有効にします。4D v11以降、マルチレベルのトランザクションがサポートされるようになりました。既存のデータベースでこの動作が考慮されていない場合、予期しない動作と なることがあるため、変換されたデータベースではこの機能がデフォルトで無効にされています (トランザクションは1レベルに制限されます)。マルチレベルのトランザクションを使用したい場合、このオプションを選択します。
**注:** このオプションはSQLエンジンで実行されるトランザクションには影響しません。SQLのトランザクションは常にマルチレベルです。
- **Unicodeモード**: カレントデータベースのUnicodeモードの有効/無効を切り替えます。Unicodeモードではデータベースエンジン、言語、メニューなどでネイティブにUnicode文字が処理されます。非Unicodeモードでは日本語環境の場合Shift_JISが使用されます。
この設定に関わらずデータファイルはUnicodeが使用され、文字列の評価にはICUが使用されます。
v2004以前から変換されたデータベースではこのオプションが選択されていません。しかしながらこのオプションを選択し、必要なコードの修正を適用することを強く推奨します。非Unicodeモードを使用しても、過去のバージョンとの完全な互換性は提供されません。
**注:**
  - このオプションのスコープはデータベースごとです。インタープリターモードではUnicodeモードのデータベースに非Unicodeモードのコンポーネントをインストールしたり、あるいはその逆を行ったりすることが可能です。
  - SET DATABASE PARAMETERコマンドを使用してUnicodeモードを設定することができます。

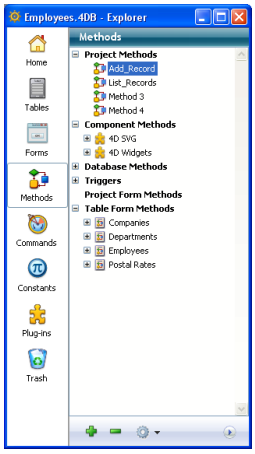
4DにおけるUnicodeサポートについてはASCIIコードを参照してください。
- **ピリオドとカンマを数値フォーマットのプレースホルダーとして使用する**: v11以降、4Dは数値の表示フォーマットにシステムの地域設定パラメーターを使用するようになりました (表示フォーマット の"数値フォーマット"参照)。4Dは自動で数値表示フォーマット中の"."を千の位区切り文字、","を小数点として解釈し、システムに設定された記号で置き換えます。以前のバージョンでは数値表示フォーマットでシステムの地域設定は考慮されていませんでした。例えば"###,##0.00"フォーマットは日本語システムでは有効ですが、フランス語システムでは結果が異なっていました。
変換されたデータベースでは互換性保持のためこの新しいメカニズムが無効になっています。
- **Web変数に値を自動的に代入する**: 以前のバージョンの4Dでは、Webサーバーの標準機構によって、HTTPフォームやGET type URL を使用して送信された変数の値を自動的に4Dプロセス変数へと代入をしていました。インタープリタモードでは、変数同士が同じ名前であれば、受け取った変 数の値はどんな値でも4Dプロセス変数へとコピーされました。コンパイルモードでは、変数は事前に*COMPILER_WEB* プロジェクトメソッドを使用して宣言しておく必要がありました。
v13.4 以降この機構は廃止予定となり、新しいデータベースでは使用できなくなりました。変換されたデータベースではこの機能は互換性のために、残されてはいます が、互換性のオプションでチェックを外すことによって無効化することができます。今後は代わりに WEB GET VARIABLES またはWEB GET BODY PART コマンドの使用が推奨されます。

# エクスプローラー


概要と動作
ホームページ
テーブルページ
フォームページ
メソッドページ
コマンドページ
定数ページ
プラグインページ
ゴミ箱ページ
コメントを使用する

## 概要と動作

エクスプローラーはデザインモードのウィンドウで、テーブルやフォーム、4Dコマンド、定数、プラグインに素早くアクセスするために使用します。またこれらの項目に関する詳細情報も提供します。**デザイン > エクスプローラー**サブメニューからページを選択するか、ツールバーのエクスプローラーボタンをクリックすることで、エクスプローラーを表示できます。




エクスプローラー左側のボタンからデザイン環境オブジェクトのそれぞれのページにアクセスできます。エクスプローラーにはテーブル、フォーム、メソッド、コマンド、定数、プラグインページがあります。
ホームとゴミ箱ページは特定の追加機能を提供するページです。

特定のページを表示すると、エクスプローラー内に対応するオブジェクトがリスト表示されます。それぞれのページでオブジェクトは階層状に表示されます。リストの下には追加や削除、オプションメニューなどのコントロールがあります。

### オブジェクトの名称変更

フォルダー、テーブル、フォーム、メソッドの名称変更が必要になったら、Alt (Windows) または Option key (Mac OS) キーを押しながらオブジェクト名をクリックします。また選択したオブジェクト上でクリックを行うこともできます (ダブルクリックにならないよう、間隔をあける必要があります)。するとテキストが編集可能になります。

変更を行い、テキストエリアの外をクリックすると変更が保存されます。

**注:** フォーム名を変更すると、その名称を以前から使用しているメソッドやフォーミュラが無効になります。正しく動作させるためにはすべての箇所を変更しなければなりません。

エクスプローラーは常にオブジェクトを文字順でリストします。新しい名称により並び順が変更される場合、4Dは入力エリアの外がクリックされたときに並び替えを行います。

**4D Server:** 名前の入力エリアの外をクリックすると、サーバー上でオブジェクト名が更新されます。同時に複数のユーザーが一つのオブジェクトの名称を変更した場合、最後に更新を行ったユーザーが付けた名称が最終的に使用されます。オブジェクトのオーナーを設定して、名称変更ができるユーザーを制限できます。アクセス権の設定に関する詳細は[ユーザー＆グループ]を参照してください。

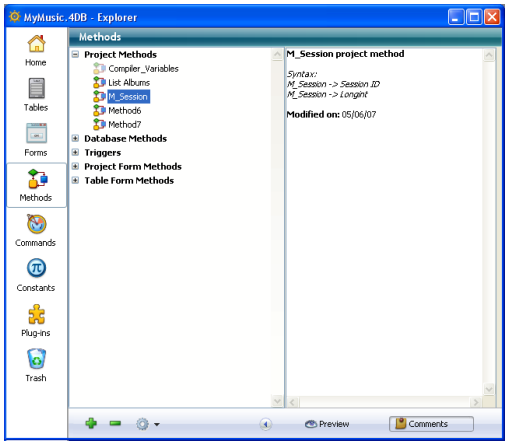
### プレビューエリアの表示と非表示

エクスプローラーのプレビューエリアを表示したり隠したりするには、展開/縮小アイコンをクリックします。プレビューウィンドウではテーブルのイメージ、フォーム、メソッド、コマンドのドキュメント、定数値、さらにデータベースにインストールされているコンポーネントやプラグインに関する情報が表示されます。またプレビューエリアには特定のデータベースオブジェクトに対し、コメントを入力することができます。
プレビューエリアを非表示にするには、再び展開/縮小アイコンをクリックします。

### コメントの表示

4Dでは以下のデータベースオブジェクトにコメントを割り当てることができます: データベースのフォルダー、テーブル、フィールド、フォーム、そしてメソッド (データベースメソッド、プロジェクトメソッド、トリガー、およびフォームメソッド)。コメントの表示や編集はエクスプローラーで行います (プレビューエリアを展開する必要があります)。

オブジェクトのコメントを表示したり更新したりするには、オブジェクトリストから目的のオブジェクトを選択し、プレビューエリアの下部に位置する**コメント**ボタンをクリックします。するとプレビューエリアはコメントエリアになります:



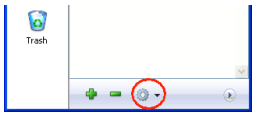

コメントの作成と利用については[コメントを使用する]で詳細に説明しています。

### ドラッグ＆ドロップの利用

多くのインスタンスで、エディターウィンドウにデータベースオブジェクトを追加するためにドラッグ＆ドロップを使用できます。例えばエクスプローラーのテーブルページからフィールド名をドラッグして、フォームエディター上に開かれたフォームにフィールドを追加できます。メソッドエディターで作業をしているときは、テーブル、フォーム、フィールド、プロジェクトメソッド、定数、コマンド、およびシンタックスなどの名前をドラッグ＆ドロップでメソッドに追加できます。
この章の後の節では、エクスプローラーのそれぞれのページについて説明すると同時に、そのページのドラッグ＆ドロップの動作についての情報も提供します。

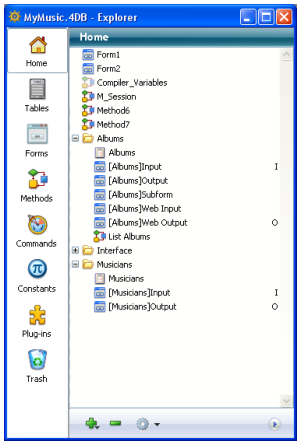
### コンテキストメニューとオプションメニューの利用

エクスプローラーのホーム、フォーム、メソッド、そしてゴミ箱ページではコンテキストメニューとオプションメニューを利用できます。これらのメニューは追加の機能へのアクセスを提供します。この点についてはこの章の後の節で説明します。



# ホームページ

ホームページでは4Dエクスローラー中のオブジェクトを管理するためのフォルダーを作成し、利用できます。




4Dのフォルダーはタイプだけではなく特定の条件、特に機能でオブジェクトをグループ化するために使用します。例えば"E-mail"という名称のフォルダーに、データベースでメールを扱うためのオブジェクトをグループ化することができます。

テーブル、フォーム、メソッドオブジェクトをフォルダーに配置できます。複数レベルのサブフォルダーを作成することもできます。

すべてのカスタマイズされたデータベースオブジェクト (例: テーブル、フォーム、プロジェクトメソッド) は常にホームページ上にも表示されます。

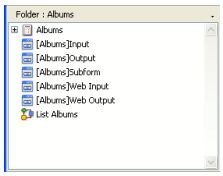
このページではフォルダーやサブフォルダーの追加、名称変更、移動、オブジェクトのフォルダーへのグループ化などができます。またテーブルやフォーム、プロジェクトメソッドを直接作成することもできます。エクスプローラーの他のページのように、名前をダブルクリックすることでテーブルやフォーム、プロジェクトメソッドを対応するエディターで開くことができます。フォルダーやオブジェクトの削除をすることもできますし、オブジェクト作成時にフォルダーを指定することもできます。

このページからも、様々なタイプのオブジェクト (メソッド、フォーム等) をデータベースの各エディターや他のデータベースのデザインモードにドラッグ＆ドロップできます。各オブジェクトタイプごとに利用可能なドラッグ＆ドロップオプションについては、このテーマの対応するページで説明しています。データベース間のドラッグ＆ドロップについてはオブジェクトのドラッグ＆ドロップで説明しています。

**4D Server:** 同じデータベースに接続するすべてのクライアントは、同じフォルダー構造を共有します。ホームページでクライアントマシンにより移動や追加などの変更が行われると、その変更は自動および即座にすべての接続されたクライアントに有効となります。

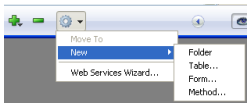
### フォルダーやサブフォルダーの作成

フォルダーはデータベースオブジェクトを機能やその他の条件 (データやランゲージ等) で整理するのに有用です。これによりエクスプローラーから異なるオブジェクトへのアクセスが容易になります。フォルダーのリストは4Dメソッドエディターのオブジェクトリストでも階層形式で表示できます。これにより、メソッド記述時に適切なオブジェクトが探しやすくなります:




またフォルダーはストラクチャーエディターで表示をフィルターする際にも使用されます (フォルダーごとにテーブルをハイライト/薄暗くする参照)。

デフォルトでデフォルトフォルダー、およびそのサブフォルダー (デフォルトテーブル、デフォルトフォーム、デフォルトプロジェクトメソッド) が作成されます。すべてのオブジェクトは作成時にトップレベルに配置されます。エクスプローラのホームページで必要な時にフォルダーやサブフォルダーを追加できます。これを行うにはページ内のオプションメニューまたはコンテキストメニューから**新規>フォルダー**コマンドを選択します:




あるいはエリア下部の追加アイコン (+) をクリックして表示されるメニューを使用することもできます。

フォルダーあるいはサブフォルダーがすぐに作成され、デフォルトで*"フォルダ_+フォルダ番号"*の名称が付けられます。この名前は必要に応じて変更できます。しかしフォルダーの名称はサブフォルダーも含めて重複することはできず、31文字までなければなりません。

- フォルダーを作成する場合、新規フォルダーを追加する際に何も項目が選択されていないことを確認してください。
- サブフォルダーを作成するには、サブフォルダーを追加するフォルダーを選択してから、新しいフォルダーを作成します。

作成したフォルダーはドラッグ＆ドロップや移動コマンドを使用して階層レベルを変更できます。

ホームページの**新規>**コマンドではフォルダーだけでなくテーブル、フォーム、プロジェクトメソッドも作成できます。このタイプのサブコマンドを選択するとき、指定されたオブジェクトを作成するためのダイアログボックスに直接移動します。

ホームページのコンテキストメニューとオプションメニューは、選択されたオブジェクトのタイプに応じて追加の機能を提供します。それぞれのタイプごとで利用可能なオプションについては、このテーマの対応する節で説明しています。

### オブジェクトの削除

**削除**コマンドは選択したオブジェクトを削除します。選択されたオブジェクトにフォルダーやサブフォルダーが含まれる場合、含まれるすべての項目も削除されます。

このコマンドはすべてのタイプの項目 (フォルダー、サブフォルダー、テーブル、フォーム、プロジェクトメソッド) とそれらの複数の選択に使用できます。

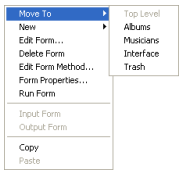
削除された項目はエクスプローラーのゴミ箱に置かれます。エクスプローラーのゴミ箱の動作についてはゴミ箱ページを参照してください。

エクスプローラーホームページからオブジェクトを削除するには、項目を選択して、コンテキストメニューあるいはページのオプションメニューから**移動>ゴミ箱**コマンドを使用することができます。

### 移動

ホームページ上のコンテキストメニュー内の**移動**コマンドを使用すると、事前に移動先のフォルダーを開かずとも、フォルダーの内容を変更できます。

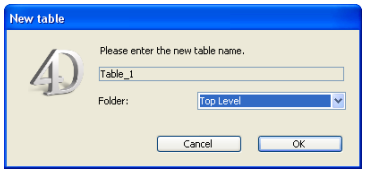
これを行うには、選択したオブジェクトやフォルダー上で右クリックを行いコンテキストメニューを表示させます。このメニュー中で**移動**コマンドのサブメニューにはすべてのフォルダーとサブフォルダーがリストされます:



選択したオブジェクトの移動先フォルダーを選択します。**トップレベル**を選択すると、オブジェクトはリストの最初のレベルに置かれます。この項目はすでにトップレベルにあるオブジェクトを移動する際には選択不可になります。**ゴミ箱**を選択すると、オブジェクトは削除されます (前段落参照)。

**作成時にオブジェクトをフォルダーに追加する**

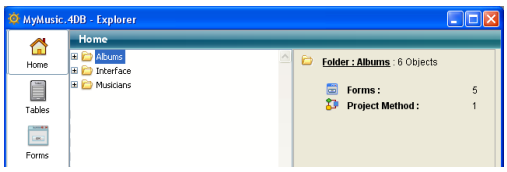
新しいオブジェクトを作成時に、直接フォルダーに置くことができます。テーブル、フォーム、プロジェクトメソッド作成ダイアログボックスのフォルダーメニューを使用して、オブジェクトを配置するフォルダーを指定できます。デフォルトでオブジェクトはトップレベルまたはエクスプローラーで最後に選択されたフォルダーに作成されます。フォルダー選択メニューは例えば以下のようにテーブル作成ダイアログボックス中にあります:




もちろんエクスプローラーのホームページでオブジェクトを作成した後に、他のフォルダーに移動することもできます。

**フォルダー情報**

エクスプローラー中でフォルダーが選択されているとき、ウィンドウのプレビューエリアにはいくつかの情報が表示されます:




- **フォルダー: フォルダー名**: フォルダー内のオブジェクト数
- **テーブル**: フォルダー内のテーブル数
- **フォーム**: フォルダー内のフォーム数
- **プロジェクトメソッド**: フォルダー内のプロジェクトメソッド数
- **フォルダー**: フォルダー内のサブフォルダー数

フォルダー内にオブジェクトがない場合、プレビューエリアに情報は表示されません。フォルダーが空の場合、エリアにはフォルダー名のみが表示されます。複数のフォルダーやオブジェクトが選択されている場合、表示される項目は、例えば最後にクリックされた、カレント項目に関する情報です。

## テーブルページ

テーブルページにはデータベース中のすべてのテーブルやフィールドがリストされます。このページをストラクチャーエディターの代わりにして、テーブルやフィールドのプロパティにアクセスできます。テーブルを展開すると、テーブル中のフィールドが表示されます。

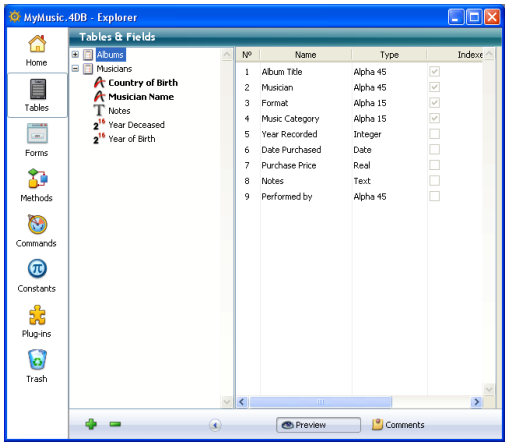
フィールドタイプはアイコンとしてフィールド名の左に表示されます。インスペクターパレットにプロパティを表示するにはフィールド名をダブルクリックします。フィールドプロパティに関する詳細はGet database parameterを参照してください。

### ドラッグ＆ドロップの使用

テーブルページのフィールド名をフォーム上にドラッグして、フォームにフィールドを追加できます。メソッドに名前をドラッグすると、テーブル名やフィールド名をメソッドに挿入できます。これを行うと、名前は正しいシンタックスで表示されます。例えば[Customers]テーブルの"First Name"フィールドをメソッドエディターにドラッグすると、メソッドに"[Customers]First Name"が挿入されます。

### プレビューエリア

エクスプローラーのプレビューエリアでテーブルの表形式で見ることもできます。これを行うにはテーブルを選択してプレビューアイコンをクリックします:




このエリアには情報が表示されるだけです。値を変更することはできません。
選択したフィールドのフィールド番号、名前、型、インデックス (タイプに関わらず)、さらにテーブルを基点とするリレーションが表示されます。リレーション列にはNフィールドに対応する1フィールドが表示されます。

リスト中でフィールドを選択すると、フィールドの型が表示されます:

### テーブルのイメージをストラクチャーウィンドウに表示する

テーブル名をダブルクリックすると、ストラクチャーエディターウィンドウでそのテーブルが表示されます。これを行うと、4Dはそのテーブルをストラクチャーエディターの中央に表示し、インスペクターパレットにプロパティを表示します。この機能は大きなストラクチャーにおいて、ストラクチャーエディターウィンドウをスクロールせずに目的のテーブルを表示させるときに便利です。

### テーブルやフィールドの追加

追加ボタンを使用してテーブルやフィールドを追加できます。

テーブルを作成するには、テーブル名を選択するかオブジェクトが全く選択されていない状態で、追加ボタンをクリックします。標準の新規テーブルダイアログボックスが表示されます (テーブルの作成と変更参照)。フィールドを作成するには、既存のフィールドを選択して追加ボタンをクリックします。インスペクターパレットが表示され、選択したテーブルのフィールドを作成することができます (参照)。

### テーブルの削除

削除ボタンを使用してテーブルを削除できます。テーブルを選択してこのボタンをクリックすると、テーブルは4Dのエディターから削除され、もう使用できなくなります。このテーブルに属するフォームやメソッドも削除されます。

**Note:** 削除ボタンを使用してフィールドを削除することはできません。

実際は、テーブルは恒久的に削除されたわけではありません。それはゴミ箱に移動され、4Dからは隠された状態です (ゴミ箱ページ参照)。ゴミ箱が空にされると、テーブルも実際に削除されます。

ストラクチャーエディター (テーブルの削除参照) またはSQLランゲージを使用してテーブルを恒久的に削除することができます。

# フォームページ

フォームページには2つのフォームリスト、**プロジェクトフォーム**と**テーブルフォーム**があります。テーブルフォームのリストにはデータベーステーブルがリストされ、さらにテーブルに属するフォームが表示されます。

プロジェクトフォームはテーブルに属さない独立したフォームです。このようなフォームはインターフェースやコンポーネントの開発に使用されます。

テーブルフォームは特定のテーブルに属するフォームです。この種類のフォームではデータ処理の際にレコードに関する自動機能や追加機能を使用することができます。これら2タイプのフォームの違いについて詳細はテーブルフォームとプロジェクトフォームを参照してください。

### 新規フォームを作成する

フォームページでは2つの方法でフォーム (プロジェクトフォームやテーブルフォーム) を作成できます:

- フォームウィザードを使用する: (プロジェクトフォームを作成する場合) "プロジェクトフォーム"タイトル、または(テーブルフォームを作成する場合) 新しいフォームを追加するテーブル (またはテーブルに属するフォーム) を選択し、ページのオプションメニューから**フォームウィザード...** を選択します:

  フォームウィザードが表示され、新しいフォームを作成する準備ができます。このウィザードの使用方法に関する詳細は、"ウィザードを使用して新しいフォームを作成する"を参照してください。
- 空のフォームを作成する: (プロジェクトフォームを作成する場合) "プロジェクトフォーム"タイトル、または(テーブルフォームを作成する場合) 新しいフォームを追加するテーブル (またはテーブルに属するフォーム) を選択し、ページ下部の追加ボタンをクリックします。フォーム名とフォルダーを指定するダイアログボックスが表示されます。ダイアログを確定すると空のフォームが作成され、フォームエディターウィンドウ中に表示されます。詳細は"エクスプローラーを使用して空のフォームを作成する"を参照してください。

### フォームの編集

エクスプローラーから直接フォームエディターに既存のフォームを開いて編集することができます。これを行うには:

- フォーム名上でダブルクリックします。
- フォームを選択し、コンテキストメニューまたはページのオプションメニューから**フォーム編集...**を選択します。

フォームの編集に関する詳細はフォームの作成の章を参照してください。

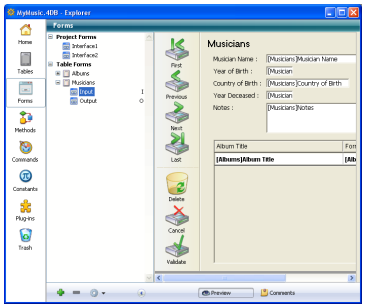
### フォームの削除

フォームを削除するには、リスト中で名前を選択し、削除ボタンをクリックするか、コンテキストメニューから**フォーム削除**コマンドを選択します。フォームエクスプローラーのをゴミ箱にドラッグ＆ドロップすることもできます。

テーブルのカレント詳細/一覧フォームを削除することはできません。

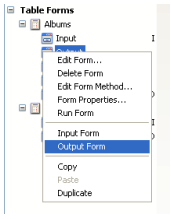
### フォームのプレビュー

プレビューエリアを表示するにはアイコンをクリックします。プレビューするにはフォームを選択します。




### カレントの詳細/一覧フォームを指定する

エクスプローラーを使用して、テーブルごとにカレントの詳細フォームと一覧フォームを指定できます (テーブルフォームのみ)。これを行うには、階層リスト中で指定するフォームを選択し、コンテキストメニューから**入力フォーム**または**出力フォーム**コマンドを選択します:




デフォルトの詳細フォームと一覧フォームについては、"詳細/一覧フォームを指定する"を参照してください。

### フォームメソッドの編集

エクスプローラーのフォームページから直接フォームメソッドを開くことができます。これを行うにはコンテキストメニューまたはオプションメニューから**フォームメソッド編集...**コマンドを選択します。フォームメソッドはメソッドエディターウィンドウ内に開かれます。

### エクスプローラーからフォームプロパティを設定する

特別なダイアログボックスを使用して、エクスプローラーから主なフォームプロパティにアクセスできます。これを行うには、指定するフォームを選択して、コンテキストメニューまたはページのオプションメニューから**フォームプロパティ...**コマンドを選択します。このダイアログボックスで設定できるプロパティに関する詳細は、フォームプロパティ (エクスプローラー)を参照してください。

### フォームを実行する

レコード表示ウィンドウ内で、そのコンテキスト (一覧テーブルフォームの場合レコードリスト、詳細テーブルフォームの場合カレントレコード) で、フォームを実行できます。これを行うにはコンテキストメニューまたはページのオプションメニューから**フォーム実行**を選択します。フォームのテストに関する詳細情報はツールバーを使用するを参照してください。

### ドラッグ＆ドロップの利用

ドラッグを使用してメソッドにフォーム名を追加できます。これを行うと、フォーム名は正しいシンタックスで挿入されます。例えば[Company]テーブルの"Input"フォームをドラッグすると、メソッドには[Company];"Input"が挿入されます。

エクスプローラーのフォームページからフォームエディターで開かれたフォームに一覧フォームの名前をドラッグすると、サブフォームを追加できます。詳細サブフォームを追加するにはShiftキーを押しながら、エクスプローラーのフォームページからフォームのサブフォームエリアに詳細フォーム名をドラッグします。詳細はサブフォームとウィジェットを参照してください。

### フォームのコピー、ペースト、複製

コンテキストメニューやページのオプションメニューを使用して、既存のフォームのコピー/ペースト、および複製ができます。これにより同じ特徴をもったフォームを複数作成することが簡単で素早くなります。

フォームの保存先に既に同じ名称のフォームが存在する場合、4Dは自動でフォーム名に番号を付加し、インクリメントします。

## メソッドページ

メソッドページにはプロジェクトメソッド、データベースメソッド、トリガー、そしてプロジェクトフォームやテーブルフォームのフォームメソッドがリストされます。これらのメソッドはカテゴリーごとにグループ化されます。

**注:** データベースに共有メソッドを含むコンポーネントがインストールされている場合、このページにはコンポーネントメソッドも表示されます。この項目にはインストールされているコンポーネントと、共有されるメソッドが表示されます。この点についてはコンポーネントの表示で説明しています。

### 新規メソッドの作成

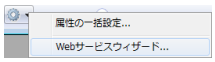
ここではエクスプローラーを使用してメソッドを新規に作成する方法をタイプごとに説明します。

- **プロジェクトメソッド**: 新規にプロジェクトメソッドを作成するには、階層リストでプロジェクトメソッド項目または既存のプロジェクトメソッドを選択し、追加ボタンをクリックします。
  複製やコピー/ペーストを使用してプロジェクトメソッドを作成することもできます (後述)。
- **データベースメソッド**: 新規にデータベースメソッドを作成することはできません。そうではなく、既存の空のデータベースメソッドにコードを追加することができます。これを行うにはデータベースメソッド項目を展開し、更新するデータベースメソッドをダブルクリックするか、コンテキストメニュあるいはオプションメニューから**メソッド編集...**を選択します。
- **トリガー**: トリガーを作成するには階層リスト中でトリガーを展開し、テーブルを選択して、追加ボタンをクリックするか、テーブル名をダブルクリックします。またコンテキストメニューあるいはオプションメニューから**メソッド編集...**を選択することもできます。
- **プロジェクトフォームメソッド**: プロジェクトフォームに新しくフォームメソッドを作成するには、階層リストのプロジェクトフォームメソッド項目を展開し、フォームを選択して、追加ボタンをクリックするか、フォーム名をダブルクリックします。またコンテキストメニューあるいはオプションメニューから**メソッド編集...**を選択することもできます。
- **テーブルフォームメソッド**: テーブルフォームに新しくフォームメソッドを作成するには、階層リストのテーブルフォームメソッド項目を展開し、テーブルを展開し、フォームを選択して、追加ボタンをクリックするか、フォーム名をダブルクリックします。またコンテキストメニューあるいはオプションメニューから**メソッド編集...**を選択することもできます。

**注:** エクスプローラーのフォームページでコンテキストメニューやオプションメニューの**フォームメソッド編集...**コマンドを選択して新規フォームメソッドを作成することもできます。

### Webサービスウィザードでメソッドを作成する

メソッドページのオプションメニューから**Webサービスウィザード...**を選択すると、Webサービスウィザードが表示されます。




Webサービスウィザードを使用してWebサービスを検索し、対応するプロクシーメソッドを作成できます。ウィザードの使用方法に関する詳細は"4DでWebサービスを使用する"を参照してください。

### メソッドの削除

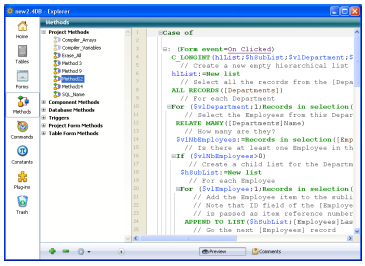
メソッドを選択して、ページ下部の削除ボタンをクリックするかコンテキストメニューから**メソッド削除...**を選択することでメソッドを削除できます。

削除の動作はメソッドのタイプによって異なります:

- データベースメソッド、トリガー、およびフォームメソッド: メソッドの内容が消去され、メソッドはデフォルトの空の状態となります。削除が恒久的であることを警告するダイアログボックスが表示されます。
- プロジェクトメソッド: メソッドはメソッドリストから削除され、ゴミ箱に置かれます。ゴミ箱からそれを取り出したり、恒久的に削除 (ゴミ箱ページ参照) したりできます。

### メソッドのプレビュー

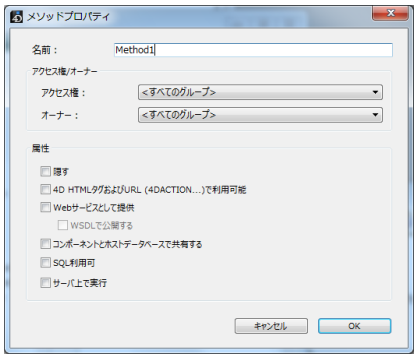
アイコンをクリックしてプレビューエリアを表示し、プレビューしたいメソッドを選択します。するとメソッドの内容はプレビューエリアに表示されます。プレビューエリアの内容全体あるいは一部をハイライトしてコピーしたりドラッグ&ドロップしたりできます。



### プロジェクトメソッドプロパティ

エクスプローラー中でプロジェクトメソッドを選択し、コンテキストメニューやオプションメニューから**メソッドプロパティ...**を選択してプロジェクトメソッドのプロパティを表示および更新することができます。

"メソッドプロパティ"ダイアログボックスが表示され、メソッド名や他のメソッドプロパティを変更できます。




このダイアログボックスの他のオプションについてはプロジェクトメソッドプロパティを参照してください。

### 属性の一括設定

コンテキストメニューやオプションメニューの**属性一括設定...**コマンドを使用すると、すべてあるいは一部のプロジェクトメソッドに対し、一回の処理で属性 (隠す、4D HTMLタグおよびURL (4DACTION) で利用可等) を設定することができます。詳細は"メソッド属性の一括設定"を参照してください。

### メソッドの実行

エクスプローラーで、コンテキストメニューあるいはオプションメニューから**メソッド実行...**を選択することでデータベースメソッドやプロジェクトメソッドを実行できます。

メソッドは標準モードで実行されます。詳細はメソッドの実行を参照してください。

### プロジェクトメソッドのコピー、ペースト、複製

エクスプローラーのメソッドリスト中でプロジェクトメソッドをコピー、ペースト、及び複製できます。この機能を使用して同じデータベース内でメソッドを複製したり、他のデータベースにペーストしたりできます。

メソッドを複製あるいはコピーするには、メソッドをリストから選択して、コンテキストメニューあるいはオプションメニューから対応するコマンドを選択します。同じコンテキストメニューから**ペースト**を選択すると、コピーしたメソッドがペーストされます。

メソッドを複製する場合や、ペースト先に同じ名前のメソッドが既に存在する場合、メソッド名に番号が付加されます (例えば*MyMethod1*)。この番号は必要に応じてインクリメントされます (*MyMethod2*, *MyMethod3*等)。

### 依存関係を検索

プロジェクトメソッドの依存関係をエクスプローラーから自動で検索できるようになりました。依存関係とはメソッドから直接あるいは間接に参照される他のプロジェクトメソッドやフォームのことです。依存関係を検索するには、エクスプローラーでメソッドを選択し、コンテキストメニューまたはこのページのオプションメニューから**依存関係を検索**を選択します。

結果ウィンドウにこのメソッドのすべての依存関係が表示されます。対応するエディターを使用して項目を開いたり、ドラッグ＆ドロップでコピーしたりできます。これによりデータベース間で特定の機能をコピーすることが簡単にできるようになります。

複数のプロジェクトメソッドを選択して**依存関係を検索**を選択することもできます。この場合これらすべてのメソッドの依存関係が結果ウィンドウに表示されます。

#### 依存関係の分析

依存するプロジェクトメソッドを検索するために、4D はメソッドコードを解析します:

- コードからの直接呼び出し
- 以下のコマンドに引数として渡されるメソッドの名前:
  - Open window (*control*/*MenuBox* 引数)
  - ON ERR CALL
  - ON EVENT CALL
  - New process
  - EXECUTE ON CLIENT
  - SET MENU ITEM METHOD
  - EXECUTE METHOD

ただし検索されるのはこれらのコマンドにテキストリテラル定数として渡されたメソッド名で、かつ実行可能で有効なメソッド名でなければなりません。

コードから依存関係を検索するために、4D は以下のコマンドに引数として渡されたフォーム名を検索します:

- Print form
- DIALOG
- FORM SET OUTPUT
- FORM SET INPUT
- PRINT SETTINGS
- FORM GET PROPERTIES
- Open form window

フォームが属するtable引数が認識されますが、明示的に指定されていなければなりません。

以下のような場合 4D は依存関係を検索できません:

```
 //検索できないケース

 //メソッド名がリテラルとして渡されていない場合
v:="myhandler"
ON ERR CALL(v)
 //テーブルポインターが使用されている場合
DIALOG(Table(1)->;$formname)
 // DEFAULT TABLE が使用されている場合
DEFAULT TABLE([myTable])
DIALOG("myForm") // "myForm" はプロジェクトフォームとして評価
 // SQL コードから FN を使用した関数呼び出し
Begin SQL
       SELECT Film_Title, {FN How_Many_Actors(ID) AS NUMERIC}
       ...
```

## コマンドページ

コマンドページには4D組み込みのコマンドがテーマごとにグループ化されて、すべて表示されます。これはメソッドエディターに表示されるコマンドリストと同じものです。

このページからコマンドのオンラインドキュメントにアクセスしたり、プレビューエリアでコマンドシンタックスに関する情報を見たりすることができます。

**オンラインドキュメントにアクセスする**

コマンド名をダブルクリックすると、直接4Dのオンラインドキュメント (HTML) にアクセスできます。対応するHTMLページはデフォルトWebブラウザーに表示されます。

HTMLドキュメントはCD-ROM、DVD、ハードディスクなどのメディア上、あるいは4D社のWebサイトに配置されたものを使用できます。どこからページをロードするかは環境設定の"ドキュメントの場所"で指定します (一般ページ参照)。デフォルトで4Dは4D Doc Center Webサイトに接続します。

**コマンドシンタックスの表示**

プレビューエリアには選択されたコマンドの番号とコマンドシンタックスが表示されます。

**注:** コマンド番号はCommand name関数で使用されます。

シンタックスを表示したくない場合、コマンド名をクリックするときに**Alt** (Windows) または**Option** (Mac OS) キーを押します。

エクスプローラーのリストエリアにもコマンド番号とコマンドシンタックスがヘルプTipとして表示されます:

これらのヘルプTipはメソッドエディターの編集エリアに表示されるものと同じです。

**ドラッグ & ドロップを使用する**

エクスプローラーからメソッドエディターにコマンドをドラッグすることで、メソッドにコマンドを挿入できます。

この時コマンド名とそのシンタックスも挿入されます。

Windowsではコマンドドラッグ時に**Ctrl**キーを押す必要があります。

## 定数ページ

定数ページには、メソッド中で使用できるすべての定数が階層リスト中に表示されます。選択した定数の値はプレビューエリア、およびリストエリアのヘルプTipとして表示されます。

表示される定数は4D、プラグイン、あるいはカスタムリソースから読み込まれます。メソッド中の定数に関する詳細は4Dランゲージリファレンスの定数を参照してください。

**ドラッグ＆ドロップの利用**

定数はメソッド中で頻繁に使用されます。定数をタイプする代わりに、エクスプローラー中で定数を選択し、メソッドエディターウィンドウに (Windowsの場合**Ctrl**キーを押しながら) ドラッグすることでメソッドに定数を挿入できます。デフォルトで定数は、メソッドエディターがコードを解析した後に下線付きで表示されます。

## プラグインページ

プラグインページにはデータベースにインストールされているプラグインと、そのコマンドがリストされます。通常コマンドはプラグイン内でテーマごとにグループ化されます。

プレビューエリアにはプラグインに関する追加の情報と、有効なファイルの場所が表示されます。

**注：** アプリケーションにプラグインをインストールする方法についてはSet user propertiesを参照してください。

## ゴミ箱ページ

エクスプローラーのゴミ箱ページには、データベースから削除されたフォルダー、テーブル、フォーム、プロジェクトメソッドオブジェクトが集められています。

WindowsやMac OSのゴミ箱のように、4Dのゴミ箱も事故で項目を削除してしまうリストを低減させるインターフェースです。

ゴミ箱に置かれたオブジェクトは4Dのエディターやメニューには表示されず、編集や利用、移動等はできません。同じ名称で同じタイプのオブジェクトを新規に作成することもできます。それにもかかわらず、ゴミ箱を空にしない限り、ゴミ箱内のオブジェクトを復旧させることが可能です。

ゴミ箱ページのプレビューエリアにはオブジェクトが表示されます。さらに削除されたオブジェクトに書かれたコメントも見ることができます。

**4D Server:** ゴミ箱の内容はすべてのクライアントマシンで同じです。

### オブジェクトの削除

オブジェクトはホームページ (フォルダー)、テーブル、フォーム、メソッドページから削除されます。

エクスプローラーからオブジェクトを削除するには、オブジェクトを選択し、エクスプローラーのコンテキストメニューから**削除**コマンドを選択するか、ゴミ箱にドラッグするか、ページ下部の削除アイコンをクリックします。

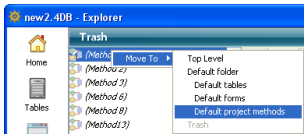
ホームページ上では、**移動 >** コンテキストメニューからゴミ箱を移動先に選択できます。

テーブルもゴミ箱に置くことができます。テーブルはストラクチャーエディターを含むデータベースのエディターに表示されなくなります。テーブルをゴミ箱に置くと、テーブルのフォームも自動でゴミ箱に移動され、割り当てられたメソッドは削除されます。

以下の属性を持つフォームは削除できません: I (カレントの詳細フォーム), O (カレントの一覧フォーム) または B (詳細/一覧属性を持つフォーム)、ただしこのフォームのテーブルが削除された場合を除きます。同様に削除ができないオブジェクトを含むフォルダーも削除することができません。

### オブジェクトの取り出し

ゴミ箱を空にしない限り、ゴミ箱ないのオブジェクトはいつでも取り出して再利用することができます。これを行うにはオブジェクトを選択し、ウィンドウ左の対応するボタンアイコン上にドラッグするか、コンテキストメニューから**移動 >** コマンドを選択します:




**移動 >** サブメニューにはデータベースに作成されたフォルダーがリストされます (詳細はホームページを参照してください)。データベースにフォルダーが作成されていない場合、**トップレベル**項目のみが利用できます。このコマンドでサブフォルダーを選択すると、選択したオブジェクトは復旧され、選択されたフォルダーに配置されます。**トップレベル**を選択すると、オブジェクトはフォルダーではなくリストのトップレベルに配置されます。

復元したいオブジェクトと同じ名前を持つ同じ型のオブジェクトがデータベース中に存在する場合、4Dは以下の選択肢を含むダイアログを表示します:

- データベース中のオブジェクトを置換する (フォームやメソッド等)
- 復元するオブジェクトの名称を*name_1*のように変更する (テーブル)

**はい**ボタンをクリックすると、復旧されるオブジェクトの名称が変更されるか、既存のオブジェクトを置き換えます。同様に**すべてはい**ボタンをクリックすると、復旧されるすべてのオブジェクトの名称が変更されるか、既存のオブジェクトを置き換えます。**いいえ**や**すべていいえ**をクリックすると、オブジェクトは復旧されません。
指定されたオブジェクトのみが復旧されます。特にテーブルの削除に伴って削除されたフォームは、テーブル復旧時に自動では復旧されません。

### ゴミ箱を空にする

デフォルトで、ゴミ箱に入れられたオブジェクトはゴミ箱が空にされるまで残ります。いつゴミ箱を空にするかは開発者が決定します。例えばストラクチャーファイルのサイズを減らすために圧縮を行う前などです。空にするとゴミ箱の中のオブジェクトは恒久的にデータベースから削除されます。

ゴミ箱を空にするにはエクスプローラのコンテキストメニューまたはオプションメニューから**ゴミ箱を空にする**コマンドを選択します:

ゴミ箱内の削除可能なオブジェクト数が表示され、処理を続行するかキャンセルするかをたずねるダイアログボックスが表示されます:

ゴミ箱内のオブジェクトをすべて削除するには**OK**をクリックします。

## コメントを使用する

エクスプローラーではデータベース中のオブジェクトに関するコメントを書くことができます。コメントの利用は特に複数のプログラマーが開発に携わっている場合に必要であり、一般的によいプログラミング習慣とされています。コメントはエクスプローラーのプレビューエリアに表示されます。

以下のオブジェクトにコメントを書くことができます:

- フォルダー
- メソッド (データベースメソッド、コンポーネントメソッド、プロジェクトメソッド、フォームメソッド、トリガー)
- フォーム
- テーブル
- フィールド
- プラグイン

コメントにはスタイル (書体や色) を付けることができます。コメントにはデータベース中でどのような役割や機能を果たすかなどのオブジェクトの説明を記述します。作成したコメントはストラクチャーファイルに保存されます。

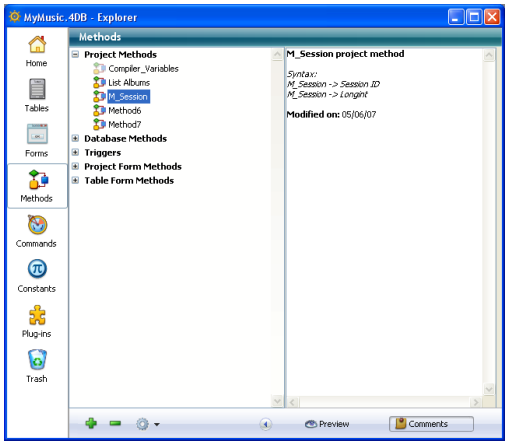
さらに4Dの**自動コメント**の機能を使用して、オブジェクトが作成/更新されたときに自動で4Dにコメントを記述させることもできます。

**互換性に関する注意**: 4D v12より、メソッドが他のメソッドで参照されたとき、// 文字を使用してメソッドのヘッダーに挿入されたコメントが、ヘルプTipとして表示されるようになりました (WA SET EXTERNAL LINKS FILTERSの"ヘルプTipを使用する"を参照)。ユーザーメソッドのドキュメント化に特に利用価値のあるこの機能は、エクスプローラーに書かれたコメントと互換性がありません。この方法でのメソッドヘッダーを使用したい場合、エクスプローラーにコメントを入力しないでください。

### オブジェクトにコメントを割り当てる

エクスプローラーを使用してコメントを作成、表示、編集します。コメントを追加するには、プレビューエリアが表示されているときに、得句ぶローラーのリストからオブジェクトを選択して、プレビューエリアの下部にある**コメント**ボタンをクリックします。

コメントオプションが選択されると、プレビューエリアはコメントエリアとなり、テキストの入力や表示ができるようになります。



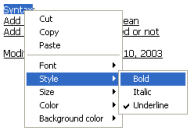

**4D Server**: エリアの左下にはロックアイコンがあり、コメントが他のユーザーによる編集されているかどうかを示します。この場合鉛筆にスラッシュがつき、コメントは見ることだけができます。

テキストは入力エリアの外をクリックすると保存されます。オブジェクトごとに32,700文字を入力できます。コメントエリア内で標準のテキスト編集コマンド (コピー、ペースト、すべてを選択等) をコンテキストメニューあるいはキーボードショートカットから使用できます。コメントエリア内で矢印キーを使用してカーソルを移動することもできます。

### コメントのスタイルを変更する

コンテキストメニューを使用して、コメントのフォント属性 (フォント、スタイル、サイズ、カラー) を変更できます。コメントテキストのスタイルを変更するには:

1. コメントエリア内で変更したいコメントテキストを選択する。
2. エリア内を右クリックする。
   階層ポップアップメニューが表示されます:




3. テキストに敷将する属性を選択する。

### 自動コメントを挿入する

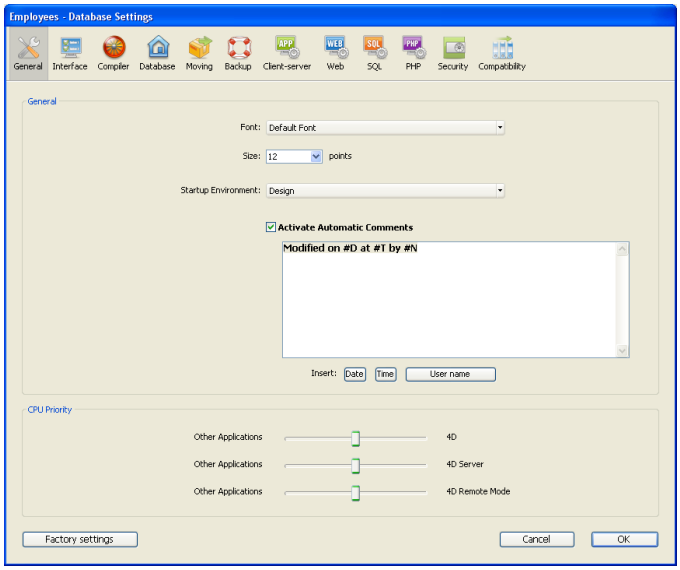
データベースのメソッドやフォームに対して、自動コメントを有効にすることができます。このシステムが有効にされると、すべてのメソッドやフォームが作成、更新されるたびにコメントが自動で割り当てられます。自動コメントにはスタティックテキスト ("更新日"のような文字列) や変数 (今日の日付、現在時刻、カレントユーザー) を使用できます。

自動コメントはデータベース設定の一般ページで設定します (参照)。標準のコメントのように、エクスプローラーで見ることができます。

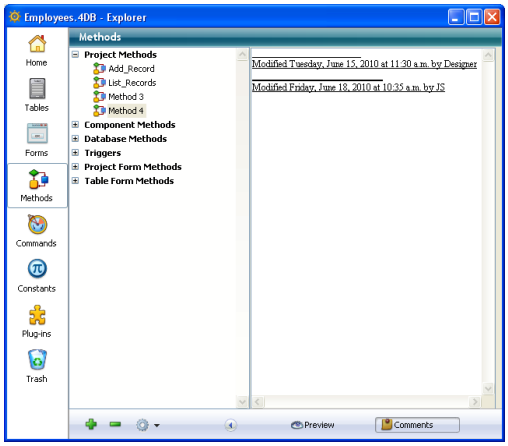
自動コメントを有効にするには以下の手順を行います:

1. **デザイン**メニューから**データベース設定...**コマンドを選択するか、4Dツールバーの**設定**ボタンをクリックします。
2. 一般ページで、**自動コメントを使用する**オプションにチェックします。
3. コメントとして設定するテキストを入力します。
   スタティックテキストおよび、エリアの下にある**日付**、**時間**、**ユーザー名**ボタンを使用して変数要素を挿入できます。以下の変数を直接入力することもできます:
   **#D** : 日付
   **#H** : 時間
   **#N** : ユーザー名

例えば以下のように入力すると:




すべての新規フォームやメソッド、および自動コメントが有効にされた後に編集されたすべてのフォームやメソッドに、自動コメントが追加されます:




**注:** パスワードシステムがデータベースで有効になっていない場合、#Nは"Designer"を返します。

**4D Server:** 自動コメントはデータベース設定にアクセスできるすべてのクライアントワークステーションが変更できます。またサーバーマシン上のデータベース設定を使用して編集することもできます。自動コメントに対して行われた変更は、それぞれのクライアントワークステーションに対し、オブジェクトが変更されコメントが受け入れられると即座に効果を持ちます。

## ランタイムエクスプローラー


ランタイムエクスプローラーを開く

ウォッチページ

プロセスページ

ブレークページ

キャッチページ

## ランタイムエクスプローラーを開く

ランタイムエクスプローラーではデータベースの様々な要素の振る舞いを表示したり、利用可能なリソースが期待通りに動作しているかなどを確認することができます。ランタイムエクスプローラーは特にデータベースの開発や解析フェーズで役に立ちます。

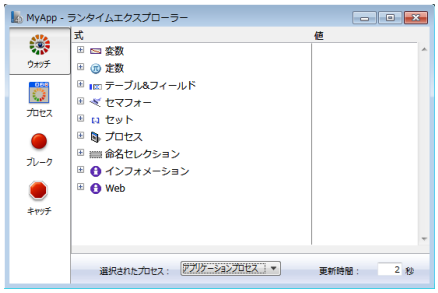
ランタイムエクスプローラーはデザインおよびアプリケーションモードで、コンパイルされていてもいなくても表示できます。

パスワード保護されたデータベースでは[デザイナーと管理者](#)だけがランタイムエクスプローラーにアクセスできます。

ランタイムエクスプローラーは標準のウィンドウあるいはフローティングパレット上に表示できます。フローティングパレットは常に他のウィンドウより前に表示されます。

- 標準ウィンドウにランタイムエクスプローラーを表示するには**実行**メニューから**ランタイムエクスプローラー**を選択します。
- フローティングパレットににランタイムエクスプローラーを表示するには:
  - Windowsでは**Ctrl+Shift+F9**、Mac OSでは**Command+Shift+F9**を押す、または
  - **Shift**キーを押しながら**実行**メニューから**ランタイムエクスプローラー**を選択します。

ランタイムエクスプローラーウィンドウには4つのページ (ウォッチ、プロセス、ブレーク、キャッチ) があり、ウィンドウ左のボタンを使用して切り替えます。



## ウォッチページ

**ウォッチ** ページはデバッガーであり、アプリケーションや選択されたプロセスに関する実行コードの情報を表示します。

ウィンドウの下部エリアでは表示する情報の種類を選択します:

- **選択されたプロセス**: このドロップダウンリストには実行中のプロセスがリストアップされます。この中から観察したいプロセスを選択します。
- **更新時間**: このエリアでは秒単位で値を入力して、ページの情報更新間隔を指定できます。

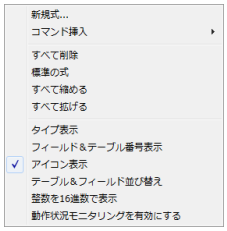
"式"列にはオブジェクトや式の名前が表示されます。"値"列にはオブジェクトや式の現在値が表示されます。これらの列は幅を変更することができます。(操作が許可されている場合) 右の列中で値をクリックすると入力が可能となり、値を変更することができます。
階層リストはテーマごとに分けられています。以下のテーマがあります:

- **変数**: データベースのインタープロセス変数や、選択したプロセスのプロセス変数のリストを表示します。
- **定数**: データベース中に定義された定数のリストを表示します。
- **テーブル＆フィールド**、**セマフォー**、**セット**、**プロセス**そして**命名セレクション**: このテーマに表示される情報は4Dのデバッガーに表示される情報と同じです。4Dランゲージリファレンスのウォッチエリアを参照してください。
- **インフォメーション**: このテーマにはデータベースの動作に関わる、カレントデフォルトテーブルや物理・仮想・空き・使用メモリ、クエリの格納先等、一般的な情報が表示されます。この情報を使用してデータベースの動作を検査できます。
- **Web**: アプリケーションのWebサーバーに関する情報を表示します (Webサーバーが有効であるときにのみ利用可能)。
  - **送出 Web ファイル**: 送信待ちWebファイル名 (存在する場合)
  - **Webキャッシュ使用**: Webキャッシュ中に存在するページ数とその利用率
  - **Webサーバー経過時間**: Webサーバーの起動時間 (HH:MM:SSフォーマット)
  - **Webヒット数**: Webサーバーが起動されて以降受信した総HTTPリクエスト数および秒あたりのリクエスト数
  - **動作中のWebプロセス数**: アクティブなWebプロセス数

  このテーマの式は変更できません。

式やテーマを削除するには、対応する行を選択し**Delete**または**Backspace**キーを押します。

またコンテキストメニューを使用してアクションを実行することもできます:




新しい式や4Dコマンドを追加したり、すべてを削除、すべての標準式を表示、すべてを折りたたみ、すべてを展開などの動作を行えます。

**注:** 式列をダブルクリックして新しい式を追加することもできます。

さらにコンテキストメニューからいくつかの表示モードにアクセスすることもできます:

- **型を表示**: テーブル＆フィールドリストにフィールドのデータ型を表示するかどうかを指定します。
- **フィールド/テーブル番号を表示**: テーブル＆フィールドリストにテーブル/フィールド番号を表示するかどうかを指定します。フィールドの場合[テーブル番号]フィールド番号のように表示されます。
- **アイコンを表示**: 階層リスト中にオブジェクトアイコンを表示するかどうかを指定します。
- **テーブル/フィールドをソート**: テーブルやフィールドを文字順にソートするかどうかを指定します (デフォルトではテーブル番号/フィールド番号順にソート)。
- **数値を16新で表示**: 整数や倍長整数型の値を16進表記するかどうかを指定します。
- **動作状況モニタリングを有効にする** (v12): スケジューラーやネットワーク通信に関連する追加情報を表示します。この低レベルな情報は**スケジューラー**と**ネットワーク項目**にグループ化されていて、アプリケーションの内部的なアクティビティの詳細なモニターを可能にします。このオプションを有効にすると処理が遅くなることに留意してください。

# プロセスページ

**プロセス**ページでは消費CPU時間やデータベースが作成するプロセスの様々な情報を表示できます。プロセス名の左にあるボタン (+/-) をクリックしてプロセスごとに表示をするかしないか切り替えられます。プロセスに関する詳細は4Dランゲージリファレンスマニュアルのプロセスを参照してください。

**4D Serverに関する注記**: 4Dクライアントのランタイムエクスプローラーの**プロセス**ページにはこのクライアントのプロセスが表示されます。4D Server管理ウィンドウのプロセスページを使用すればサーバーに接続しているすべてのクライアントのプロセスを制御できます。

## プロセスに関する情報

各プロセスごとに以下の情報が表示されます:

- プロセス番号 (コマンドでプロセスを指定するために使用される番号)
- プロセス名
- プロセスの現在の状態
- プロセスが開始されて以降の実行時間 (秒)
- プロセスのステータスおよびプロセスに割り当てられたCPU時間のグラフ表示。プロセス名の左にある (+/-) ボタンをクリックして、プロセスごとに表示を切り替えることができます。またアイコン上で**Alt+クリック** (Windows) または **Option+クリック** (Mac OS) を行うと、一度に表示/非表示を切り替えられます。

### プロセス番号

各プロセスには作成順にユニークな番号が割り振られていて、その番号は括弧内に表示されています。アプリケーション開始時に4Dが起動するプロセスに最初の番号が割り当てられます。使用される数は起動時に実行されるサーバー機能の数によって変化します。

カスタムプロセスを開始すると、プロセスリストの最後に追加されるか、アポート済みの番号が割り当てられます。例えばプロセス番号8までが使用中で、プロセス番号7がアポートされていた場合、次に開始されるプロセスの番号は7になります。

**注:** (Webプロセスなど一部を除き) 処理が終了するとプロセスは自動で終了されます。ランタイムエクスプローラーを使用すると、処理終了前でもプロセスをアポートできます。プロセスのアポートについてはプロセスをアポートを参照してください。

### プロセス名

New processやExecute on serverコマンドを使用して新規プロセスを開始する際、プロセス名を指定することができます。引数として指定されたプロセス名はプロセスリストに表示されます。

コマンドにプロセス名を指定しなかった場合、4Dがデフォルト名を割り当てます。使用されるデフォルト名は、プロセスが開始された方法に基づき以下のとおりとなります:

- **メニューから開始されたプロセス**: デフォルト名は"ML_プロセス番号"です (例えば"ML_7")。
- **メソッド実行ダイアログやメソッドエディターから開始されたプロセス**: デフォルト名は"P_プロセス番号"です (例えば"P_7")。
- **プロセス名を指定せずにコマンドから開始されたプロセス**: プロセス名は空です。

**4D Serverに関する注記:** プロセス名が"$"から始まる場合、それはローカルプロセスであり4D Server上のオブジェクトにアクセスする事はできません (グローバルプロセスとローカルプロセス参照)。

### プロセスの状態

プロセスの状態とは、プロセスが実際行っていることを示す現在の実行状況です。ランタイムエクスプローラー中でプロセスの状態はプロセス名の左側にアイコンとして表示され、さらにプロセス名の右側にテキストでも表示されます。

ランタイムエクスプローラに表示される各プロセス状態の説明は以下の通りです:

- **Executing**: プロセスは現在実行中です。
- **Delayed**: プロセスは指定された時間だけ遅延されています。プロセスが遅延されている間は処理時間を消費しません。プロセスを遅延させる方法については4Dランゲージリファレンスマニュアルを参照してください。
- **Waiting Event**: プロセスはユーザーのアクション (ボタンのクリックやメニューの選択など) を待っています。
- **Waiting I/O**: プロセスは何らかの入出力が発生するのを待っています。例えばレコードがディスクに保存されるのを待つなどの場合です。
- **Waiting Semaphore**: プロセスは、内部プロセスが4Dデータベースタスクの実行を終了するのを待っています。
- **Paused**: プロセスは実行の再開が指示されるまで一時停止しています。プロセスが一時停止されている間は処理時間を消費しません。詳細はプロセスの一時停止と再開を参照してください。
- **Aborted**: プロセスは既に終了されています。プロセスがアポートされると4Dはロックされたレコードを解除、そのプロセスがまだ開いているすべてのトランザクションをキャンセル、そしてカレントセレクションやカレントレコードを解放します。プロセスは処理が終了すると自動でアポートされます。ランタイムエクスプローラーを使用して、処理完了前にアポートすることも可能です (プロセスをアポート参照)。
- **Hidden Modal Dialog**: モーダルダイアログボックスを表示しているプロセスが非表示になっているため、ユーザーはそのダイアログボックスを見ることができません。プロセスはダイアログが表示されるまでこの状態のままです。

### 処理時間

プロセスの管理において、4Dは既存のプロセスに対し順番に処理時間を割り当てます。そのため処理時間とは、プロセスが起動されてから、そのプロセスが使用した総時間 (秒) のことであり、プロセス開始時からの経過時間ではありません。

ランタイムエクスプローラーにはプロセスごとの処理時間が表示されます。グラフを展開すると以下の情報が表示されます:

ウィンドウ下部の"更新周期"メニューを使用して描画の更新周期を切り替えることができます。高周期で描画を更新すると、それだけランタイムエクスプローラーが消費するCPU時間が増えます。画面上にグラフ表示するプロセスの数も、ランタイムエクスプローラーが消費するCPU時間に影響します。

**注:** グラフ表示しないプロセスはCPU時間を消費しません。

グラフエリアをクリックすると、その場所に縦の線が表示され、その瞬間のプロセスの状態がTipとして表示されます。マウスボタンを押しながら縦棒を左右に移動するとプロセス状態が変化する様子を見ることができます。

## プロセスに対して行える処理

ランタイムエクスプローラーからプロセスを一時停止、再開、アポートできます。またプロセスをデバッグモードにすることもできます。さらに選択したプロセスを非表示にしたり、再表示したりできます。

**注:** コマンドを使用してプロセスを指定時間だけ遅延させることもできます。

### プロセスの一時停止と再開

プロセスの実行を一時的に停止させることができます。あるプロセスを一時停止することで他のプロセスにより多くの実行時間を割り当てたり、他のイベントを割り込ませたりすることができます。

例えばレコードセレクションを印刷するプロセスを開始したとします。しかしその中のレコードを変更する必要が生じました。そこでまずプロセスを一時停止し、レコードを変更したら、印刷プロセスを再開します。

- プロセスを一時停止するには対象プロセスを選択して**一時停止**ボタン をクリックします。ランタイムエクスプローラー中でそのプロセスのステータスは自動的に"paused"に変更されます。明示的に再開が指示されるまでそのプロセスは一時停止されたままです。
- プロセス実行を再開するには対象プロセスを選択して**再開**ボタン をクリックします。プロセスの状態は一時停止される直前の状態に戻ります。例えば一時停止される前そのプロセスの状態が"Executing"であった場合、プロセスは"Executing"状態で再開されます。プロセスがイベント待ちだった場合、再開後に又イベントを待ちます。

### プロセスをアポート

プロセスは処理が終了すると自動でアポートされます。しかしデバッグの目的で、処理の終了を待たずにプロセスをアポートしたい場合があるかもしれません。それ以外の理由ではいかなる場合でもアポートすべきではありません。処理の実行を停止するには、ランタイムエクスプローラー内でプロセスをアポートします。

プロセスがアポートされると4Dはレコードのロックを解除し、そのプロセス内で開始され閉じられていないトランザクションをキャンセルし、カレントセレクションやカレントレコードを解放します。

プロセスをアポートするには対象プロセスを選択し**アポート**ボタン をクリックします。ランタイムエクスプローラー中のプロセス状態は自動で"aborted"になります。

### プロセスをトレース

4Dデバッガーを使用してプロセスの実行をモニターし、デバッグを行うことができます。

プロセスをデバッグモードにするには対象プロセスを選択し**トレース**ボタン をクリックします。

プロセスが実行中であれば4Dのデバッグウィンドウが表示され、実行をステップしながらメソッド中で使用される式の値を確認し、デバッグを行うことができます。プロセスが一時停止中である場合、4Dはデバッグリクエストを記憶します。そのプロセスの実行が再開され次第デバッグウィンドウが表示されます。

デバッグウィンドウの使用方法については4Dランゲージリファレンスマニュアルを参照してください。

4Dが作成し管理する内部プロセスをトレースすることはできません。

#### プロセスを非表示にする

アプリケーション環境においてプロセスを非表示にすることができます。プロセスが非表示にされるとそのプロセスで実行中のメニューやダイアログはユーザーに対し表示されなくなります。

プロセスの非表示は、後で再利用したいウィンドウを一時的に隠したい場合に有用です。ウィンドウを閉じるためにプロセスをアボートする代わりに、そのプロセスを非表示にすることができます。ウィンドウが隠されていてもプロセスは存在し続けます。

プロセスを隠すにはそのプロセスを選択し、オプションメニューから**隠す**を選択します:

結果プロセスはアプリケーション環境に表示されなくなります。隠されている間も実行は継続される点に留意してください。
画したプロセスをいつでも再表示できます。

プロセスを再表示するにはオプションメニューから**表示**を選択します。するとそのプロセスはアプリケーション環境に再表示されます。

#### プロセスを最前面にする

プロセス内で開かれているウィンドウを最前面にするには、そのプロセスを最前面にします。例えばアプリケーションプロセスを最前面にすると、アプリケーション環境がスクリーンの最前面となります。

どのユーザープロセスでも最前面にできます。プロセス内でウィンドウを開いている場合、そのウィンドウが最前面になります。ウィンドウにメニューバーが割り当てられていればそのメニューバーがカレントとなります。

プロセスを最前面にするにはそのプロセスを選択し、オプションメニューから**前面に**を選択します:

そのプロセス内で開かれているウィンドウがすべて最前面となります。さらに4Dは最前面プロセスウィンドウのメニューバーを表示します。

## ブレークページ

**ブレーク**ページではデータベースに設定されたブレークポイントを表示し管理することができます。

このページについては4Dランゲージリファレンスの[ブレークリスト](#)で説明しています。

## キャッチページ

**キャッチ**ページにはデータベース中でコマンドや式に対して設定されたブレークポイントが表示されます。

このページについては4Dランゲージリファレンスの[コマンドのキャッチ]で説明しています。

# デザインモードの検索と置換


概要

検索を実行する

結果ウィンドウ

内容を置換

名称変更

未使用の要素を検索する

## 概要

4Dはデザインモード全体を検索/置換する機能を複数提供します。

- カスタム条件 (前方一致や含む) に基づき、データベース全体やデータベースの一部を対象に、文字列やオブジェクトのタイプ (変数、コメント、式等) を検索できます。例えば名前が"HR_"から始まるメソッドを対象に、"MyVar"を名前に含む変数を検索するといったことができます。
- 結果は結果ウィンドウに表示されます。このウィンドウでは内容の置換等が行えます。
- コードで使用されていない変数やメソッドを検索し、メモリー節約のためにそれらを取り除くことができます。
- 一回の操作で、デザインモード全体に対し、プロジェクトメソッドや変数の名称を変更することができます。

**注:** データベースのメソッド間で検索を行う機能もあります。この機能にはエクスプローラーのメソッドページのコンテキストメニューからアクセスします。**呼び出し元を検索**と**依存関係を検索**、両方とも検索した結果は結果ウィンドウに表示します。

**検索の場所**

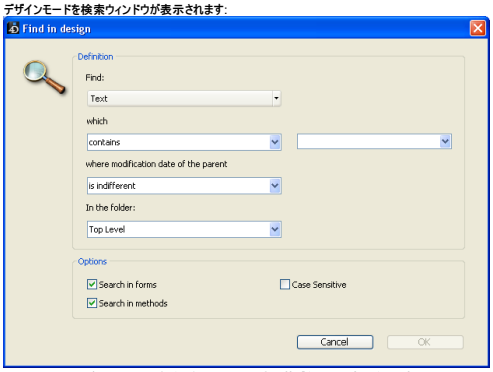
デザインモードを検索する際、デフォルトで以下のオブジェクトが検索の対象となります:

- メニュー (名前と項目) およびメニュー項目に割り当てられたコマンド
- 選択リスト (名前と項目)
- ライブラリーピクチャー (名前)
- ヘルプtips (名前と内容)
- フォーマット / フィルター (名前と内容)
- テーブルとフィールド名
- フォーム名
- プロジェクトメソッド名
- エクスプローラー中のコメント
- すべてのメソッドの内容 (トリガー、データベースメソッド、プロジェクトメソッド、オブジェクトメソッド)
- フォームの内容:
  1. オブジェクト名
  2. オブジェクトに使用されているヘルプTip名
  3. オブジェクトに使用されているピクチャー名
  4. オブジェクトに使用されている変数名
  5. オブジェクトに使用されているスタイルシートの名前
  6. フォーマット文字列
  7. オブジェクトのタイトル
  8. フィールドやテーブルの参照

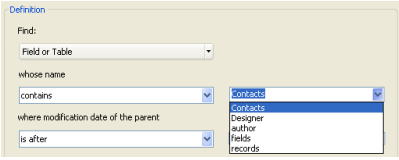
## 検索を実行する

### 検索の開始

"デザインモードを検索"ウィンドウで検索条件を指定します:

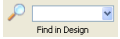
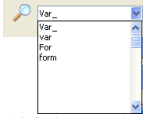
1. 4Dツールバー内の虫眼鏡ボタンをクリックするか、
   **編集**メニューから**デザインモードを検索 ...**コマンドを選択します。
   デザインモードを検索ウィンドウが表示されます:
   
   
   このウィンドウ内の各エリアは、選択されたメニューの内容に基づき動的に変更されます。
2. "定義"エリア内のメニューや入力エリアを使用して検索条件を指定し、また必要に応じて検索する文字列を入力します。
   これらの項目については後で詳しく説明します。
3. 検索オプションを設定します (必要に応じて)。
   これらのオプションについては後述の"検索オプション"で説明します。
4. **OK**をクリックするか、**Enter**キーを押します。
   検索が始まります。検索が終了すると結果ウィンドウが表示され、指定した検索条件に合致するオブジェクトがリストされます (結果ウィンドウ参照)。
   **注: Esc**キーを押すと、実行中の検索を中断できます。

検索を実行すると、入力された検索条件はメモリー内に保存されます。この値は同じセッション内でコンボボックスに表示され、選択することができます:




#### ツールバーを使用する

4Dデザインモードのツールバーを使用して検索を行うことができます。

- セッション中に検索が行われていない場合、検索したい文字列を入力して、**Carriage return** または **Enter** キーを押します。自動で"～を含むテキスト"タイプの検索がデフォルトオプションで実行されます。
  
  
- セッション中に行われた検索を繰り返したい場合、スクロールダウンリストから選択できます。
  
  
  検索が検索ウィンドウを使用して行われた場合、このウィンドウで設定されたオプションを使用して検索が実行されます。

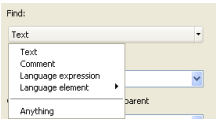
検索はすぐに始まります。検索が終了すると結果ウィンドウが表示され、入力した文字列が含まれるオブジェクトがすべてリスト表示されます。

### 検索の設定

デザインモードを検索には複数の条件を設定できます。

#### タイプ

タイプメニューを使用して、検索対象とする要素のタイプを指定できます。以下の選択肢があります:




- **テキスト:** この場合、4Dはデザインモード全体で文字列を探します。
  検索はコンテキストを考慮しないプレインテキストモードで行われます。例えば"ALERT("Error number:"+"や"button27"といった文字列を検索できます。
  このモードではワイルドカード文字を使用することはできません。"@"も通常の文字として扱われます。
- **コメント:** この検索は基本的に上のものと同じですが、検索対象はコード中およびエクスプローラーウィンドウ (コメントを使用する参照) 中のコメント (//で始まる行) に限定されます。例えば"To be verified"を含むコメントを検索できます。

**注:** 両タイプの検索結果は、検索条件メニューがどのように設定されたかにより異なります (検索モード参照)。

- **ランゲージ式:** *有効な*4D式を検索するために使用します。検索は"含む"モードで行われます。検索を行うために4Dは式を評価する必要があるため、有効な式であることが重要です。例えば"[clients" (無効な式) は結果を返しません。"[clients]"と正しい式を書かなければなりません。
  - このオプションは特に値の代入や比較を検索するのに有効です。例えば:
    "myvar:=" (代入)
    "myvar=" (比較)
- **ランゲージ要素**: 名前から特定のランゲージ要素を検索するために使用します。4Dは以下の要素を区別することができます:
  - **プロジェクトメソッド**: "M_Add"のようなプロジェクトメソッド名。(*語全体を含む*モードに割り当てられた) この検索は、メソッドエディタ内の**参照を検索**コンテキストコマンドと同等であることに留意してください (参照)。
  - **フォーム**: "Input"のようなフォーム名。このコマンドはプロジェクトフォームやテーブルフォームを検索します。
  - **フィールドまたはテーブル**: "Customers"のようなテーブルまたはフィールド名。
  - **変数**: "$myvar"の様な変数名。
  - **4D定数**: "Is Picture"のような定数。
  - **引用符内の文字列**: リテラルテキスト定数。コードエディター中の引用符に囲まれた値、またはフォームエディタのスタティックテキストエリアやグループボックスに書かれた値などです。"Martin"を検索すると、以下の行が書かれたコードが返されます:
    QUERY ([Customers];[Customers]Name="Martin")
  - **4Dコマンド**: "ALERT"のような4Dコマンド。
  - **プラグインコマンド**: アプリケーションにインストールされたプラグインコマンド。例えば"WR Find"。
- **すべて**: このオプションではデザインモードのすべてのオブジェクトを対象に検索します。更新日フィルターのみが利用できます。このオプションは"今日変更したすべてのオブジェクト"を検索する場合などに使用します。

#### 検索モード

検索モードメニュー (**検索条件**, **that is** または**名前**、検索タイプにより変化します) では入力された値でどのように検索するかを指定します。このメニューの内容はタイプメニューで選択した検索対象タイプによって変化します。

- *テキストやコメントの検索オプション*:
  - **含む**: デザインモード内で、指定されたテキストをすべて検索します。"var"を検索すると"myvar"、"variable1"、および"aVariable"が検索されます。
  - **語全体を含む**: デザインモード内で、指定された単語をすべて検索します。"var"を検索すると"var"のみが結果に含まれます。"myvar"は結果に含まれませんが、"var:=10"や"ID+var"は結果に含まれます。これは記号":"や"+"は単語区切り文字として評価されるからです。
  - **前方一致 / 後方一致**: 単語の先頭や最後 (テキスト検索)、またはコメント行の先頭や最後 (コメント検索)から、指定された文字列を検索します。"テキストの後方一致"モードでは"var"を検索すると"myvar"が見つかります。

- *ランゲージ要素の検索オプション*: 以前の4Dバージョンと同様の標準選択肢 (**等しい**, **含む**, **前方一致**, **後方一致**) が提供されます。等しいオプションではワイルドカード (@) が利用できます (指定されたタイプのすべてのオブジェクトが返されます)。

**親の更新日**

このメニューでは検索を親の作成/更新日を考慮するよう制限することができます。このメニューには標準の日付条件 (**等しい**, **より以前**, **より後**, **等しくない**) および標準の検索期間を素早く指定するための新しいオプションが含まれます:

- **本日**: 本日00:00以降の期間。
- **昨日以降**: 本日を昨日を含む期間。
- **今週**: 今週の月曜日以降。
- **今月**: 今月の一日以降。

**フォルダー**

"フォルダー"メニューは検索対象を特定のフォルダーに制限します。デフォルト ("トップレベル" オプション) ではすべてのフォルダーが検索対象となります。

**注:** オブジェクトフォルダーはエクスプローラーのホームページで定義されます。

### 検索オプション

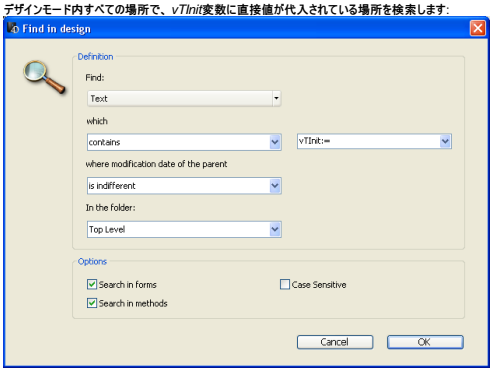
検索の速度を向上させるさまざまなオプションを選択できます。:

- **フォーム内を検索**: このオプションを選択しない場合、検索対象からフォームやフォーム名が取り除かれます。
- **メソッド内を検索**: このオプションを選択しない場合、検索対象からメソッドやメソッド名が取り除かれます。
- **大文字小文字を区別**: このオプションを選択すると、検索ダイアログに入力された文字の大文字と小文字がそのまま考慮されます。この場合"MyVar"を検索すると"myVar"は見つかりません。

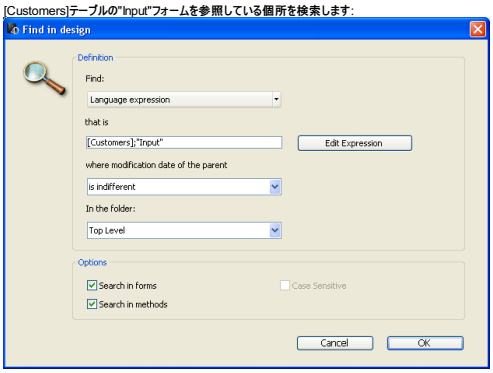
### 検索の例題

効率的に検索を行うためには、検索タイプと検索条件モードの正しい組み合わせを選択することが重要です。4D v12で検索がどのように動作するかを示すため、以下に典型的な検索の例題とどのように設定するかを示します。

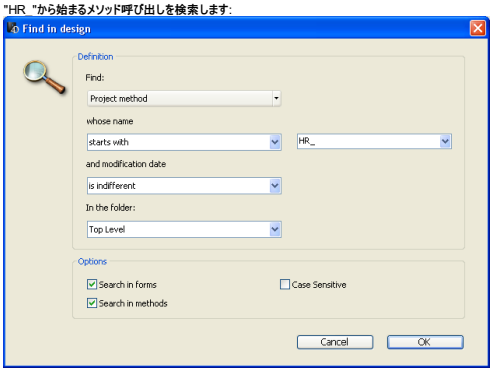
- デザインモード内すべての場所で、*vTInit*変数に直接値が代入されている場所を検索します:



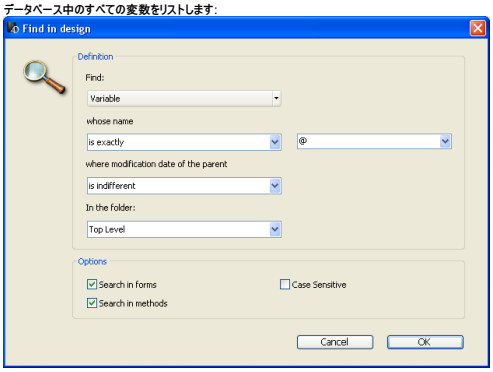

- [Customers]テーブルの"Input"フォームを参照している個所を検索します:



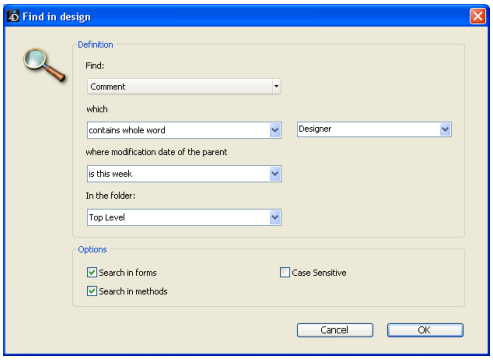

- "HR_"から始まるメソッド呼び出しを検索します:




- データベース中のすべての変数をリストします:




- 今週コメントに"Designer"キーワードが書かれた場所を検索します:

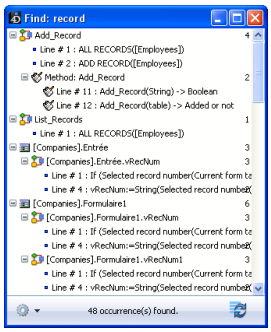

Find in design
Definition
Find:
Comment
which
contains whole word
Designer
where modification date of the parent
is this week
In the folder:
Top Level
Options
Search in forms
Case Sensitive
Search in methods
Cancel
OK

## 結果ウィンドウ

**結果ウィンドウ**には、様々な検索タイプにおいて検索条件に合致したすべてのオブジェクトが一覧表示されます:

- **標準の検索** (検索を実行する参照)
- **未使用要素の検索** (未使用の要素を検索する参照)
- **呼び出し元の検索** (呼び出し元の検索参照)
- **依存関係の検索** (依存関係を検索参照) (4D v13以降)
- **プロジェクトメソッドや変数の名称変更** (名称変更参照)

リストは検索されたオブジェクトがタイプごとにグループ化されて階層形式で表示されます。(ウィンドウの左下にある) オプションメニューやコンテキストメニュー内のコマンドを使用して階層を展開したり折りたたんだりすることができます。




このウィンドウ内の行をダブルクリックして、エディター上でオブジェクトを開くことができます。検索を複数回行うと、そのたびに結果ウィンドウが新しく表示されます。結果ウィンドウは開いたままです。

メソッド中に1つ以上のオブジェクトや文字列のオカレンスが見つかると、オブジェクト名の隣に見つかった数が表示されます。

をクリックすると、同じ条件およびオプションで再検索が行われます。

結果ウィンドウから内容を置換機能を使用することもできます。

**オプションメニュー**

オプションメニューを使用して様々なアクションを行うことができます:

**注:** これらのアクションはウィンドウのコンテキストメニューからもアクセスできます。

- **リストから削除**: 選択した項目を結果ウィンドウから取り除きます。内容を置換処理やアプリケーション間の移動対象を絞り込むために使用します。
- **内容を置換**: 選択した項目の内容の文字列を置換します。この機能については内容を置換で説明しています。
- **選択 > 結果ウィンドウ** 中の様々な項目から、特定のタイプ (プロジェクトメソッド、オブジェクト名等) の項目を選択します。階層サブメニューには一度にすべてを選択あるいは選択解除するためのコマンドも用意されています。
- **すべて折りたたむ/すべてを拡げる**: 結果リストのすべての階層項目を閉じたり展開したりします。

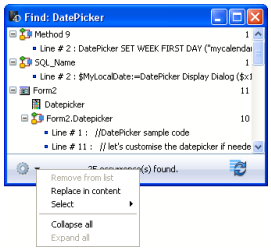
**ドラッグ＆ドロップを使用する**

**結果ウィンドウ**から項目をドラッグ＆ドロップして、2つのアプリケーション間のデザインモードでオブジェクトをコピーできます。オブジェクト移動の原則については**オブジェクトのドラッグ＆ドロップ**で説明しています。

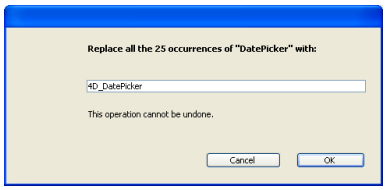
メソッド中でテーブルが参照されている場合、そのテーブルもデフォルトでメソッドとともにコピーされます。ウィンドウからメソッドをドラッグする際に**Shift**キーを押すと、この動作を無効にできます。

## 内容を置換

内容を置換は結果ウィンドウにリストされたオブジェクト内で、ある文字列を別の文字列で置き換える機能です:

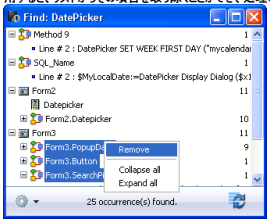



このコマンドを選択すると、検索されたすべてのオカレンスを置き換える文字列を入力するダイアログボックスが表示されます:




置換処理は以下のように動作します:

- 置換は常にリスト中のすべての項目に対し実行されます。選択された項目だけではありません。しかしコンテキストメニュー内の**削除**コマンドを使用すると、リストからその項目を取り除くことができ、処理の対象から外すことができます:

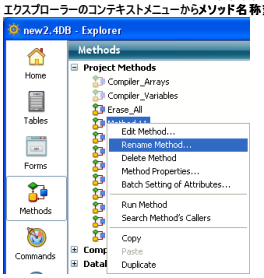



- リストに表示されているオカレンスのみが、設定された検索条件を検証した後に、置換されます。これは検索と置換の間にオブジェクトが変更された場合に備えてのことです。
- 置換は以下に対して行われます:
  - メソッドの内容
  - フォームオブジェクトのプロパティ
  - ヘルプメッセージの内容
  - 入力フィルターの内容
  - メニュー項目の内容 (項目テキストとメソッド呼び出し)
  - 選択リストの内容
  - エクスプローラーに設定されたメソッド、フォーム、テーブル、フィールドのコメントの内容
- 変更するオブジェクトごとに、4Dはそのオブジェクトが既に他のマシンや他のウィンドウでロードされていないかを検証します。衝突が発生している場合、オブジェクトがロックされている旨を示すダイアログが表示されます。オブジェクトを閉じて再実行するか、置換をキャンセルするかを決定できます。そして、置換処理はリスト中の他のオブジェクトに対して続行されます。
- "内容を置換"処理に関連するメソッドやフォームが同じ4Dアプリケーション内で現在編集中の場合、それは開かれたエディター内で直接編集されます (警告は表示されません)。この方法で変更されたフォームやメソッドは自動では保存されません。変更を有効にするには明示的に**保存**または**すべてを保存**コマンドを選択しなければなりません。
- リスト項目に置換が行われたのち、それはイタリックで表示されます。置換数はリアルタイムにウィンドウの下部に表示されます。
- フォームオブジェクトを除き、**内容を置換**機能でオブジェクトの名前は変更されません。従ってリスト中の特定の項目が置換処理の対象とならない事があります。これは検索条件に対して項目名のみが対応する場合に発生します。この場合、リスト項目はイタリックで表示されず、置換された数は検索された数よりも少なくなることになります。

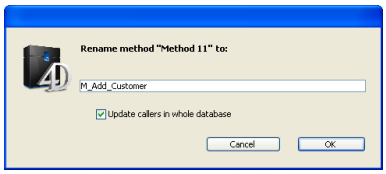
## 名称変更

4Dはデータベース全体でプロジェクトメソッドと変数の名前を変更する機能を提供しています。これは以下の方法で実行します:

- メソッドエディターのコンテキストメニューから**名称変更...**コマンドを選択する (プロジェクトメソッドと変数)。
- エクスプローラーのコンテキストメニューから**メソッド名称変更...**コマンドを選択する (プロジェクトメソッド)。




このコマンドを選択すると、オブジェクトの新しい名前を入力するダイアログボックスが表示されます:




この新しい名称は命名規則に沿っていなければなりません。そうでなければダイアログを受け入れる際に警告が表示されます。例えばメソッドに"ALERT"などコマンドと同じ名称を付けることはできません。

名称変更するオブジェクトのタイプ (プロジェクトメソッドか変数) に基づき、名称変更ダイアログボックスにはその対象を選択するオプションが含まれます:

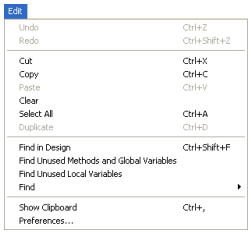
- プロジェクトメソッド: **データベース全体で呼び出し元を更新**オプションは、それを参照するすべてのデータベースオブジェクトでメソッド名を変更します。エクスプローラー内のみでメソッド名を変更するためにこのオプションの選択を外すこともできます。
- プロセスおよびインタープロセス変数: **データベース全体で変数を名称変更**オプションは、それを参照するすべてのデータベースオブジェクトで変数名を変更します。このオプションの選択を外すと、変数はカレントメソッド内でのみ変更されます。
- ローカル変数: オプションはありません。変数名はカレントメソッド内でのみ変更されます。

#### 検索の再実行

ボタンをクリックすると、同じ条件およびオプションで検索が再実行されます。これは例えば置換を行った後に、期待通りの結果が得られたかを確認するときなどに便利です。

## 未使用の要素を検索する

2つの新しいコマンドを使用して、コード中で使用されていない変数やメソッドを検出することができます。使用メモリーを減らす目的などで、これらを取り除くことができます。
このコマンドはデザインモードの**編集**メニュー内にあります:




### 未使用のメソッドとグローバル変数を検索

**未使用のメソッドとグローバル変数を検索**コマンドは、定義されているが使用されていないプロジェクトメソッドやグローバル変数 (プロセスおよびインタープロセス変数) を探します。検索結果は標準の結果ウィンドウウィンドウに表示されます。

プロジェクトメソッドは以下の場合に未使用と判断されます:

- ゴミ箱の中にない
- 4Dコード中から呼び出されていない
- メニューコマンドから呼び出されていない
- 4Dコード中で文字定数として呼び出されていない (4Dは、括弧にふくまれる引数を伴ってメソッド名が書かれていても、それを検知します)

プロセスおよびインタープロセス変数は以下の場合に未使用と判断されます:

- 4DコードでC_XXXやARRAY XXXタイプの定義コマンドを使用して定義されている
- 4Dコードの他の場所で使用されていない
- フォームオブジェクトで使用されていない

特定のケースではこの機能により検出できない場合があります。例えば実際は使用されているのに、未使用と判定される場合があります。例えば以下のようなコードの場合です:

```
v:="method"
EXECUTE FORMULA("my"+v+String(42))
```

このコードはメソッド名を計算で構築しています。この場合mymethod42プロジェクトメソッドは、実際に呼び出されてはいますが、未使用と判定されます。
そのため、未使用と判定されたオブジェクトを削除する前に、その要素が本当に必要ないかを検証されることをお勧めします。

### 未使用のローカル変数を検索

**未使用のローカル変数を検索**コマンドは定義されているが使用されていないローカル変数を検索します。検索結果は標準の結果ウィンドウに表示されます。

ローカル変数は以下の場合に未使用と判定されます:

- 4Dコード中でC_XXXやARRAY XXXタイプのコマンドを使用して定義されている
- 同じメソッド内の他の場所で使用されていない

## オブジェクトのドラッグ＆ドロップ


概要

移動ダイアログボックス

移動プロパティ

## 概要

4Dのデザインモードでは、2つのアプリケーション間でオブジェクトを移動させることができます。あるデータベースで作成したテーブルやフォーム、メソッドを他のデータベースにコピーできます。これによりデータベースの開発がより容易になります。

オブジェクトの移動は個々のオブジェクトに制限されません。それらのオブジェクトから参照される、例えば依存するオブジェクトも関連します。そのため、機能全体を移動することができます。例えばカスタム検索ダイアログを作成した場合、ダイアログボックスとして使用するフォーム、並びに使用されるすべてのメソッド、ピクチャー、その他のオブジェクトを移動できます。フォームを他のデータベースまたはデータベースで現在使用されている機能をグループ化するライブラリーにコピーできます。
特定のオブジェクトもまた分割できません。それらは必ず親オブジェクトとともに移動されます。分割できないオブジェクトに関しては後述します。

### 移動可能なオブジェクト

オブジェクトはツールボックス、エクスプローラー、そしてフォームエディターから移動できます。さらにデザインモードを検索の結果ウィンドウから移動を実行することもできます (結果ウィンドウ参照)。

構造上の整合性を保つため、特定のオブジェクトをコピーする場合に、そのオブジェクトと不可分のオブジェクトもコピーされることがあります。例えばフォームをコピーすると、フォームメソッドやオブジェクトメソッドもコピーされます。これらの分割できないオブジェクトはそれ単体では移動できません。

以下は移動可能なオブジェクトと、それと分割できないオブジェクトのリストです:

| 移動可能なオブジェクト | 分割できないオブジェクト |
|---|---|
| **---- ツールボックス ----** | |
| リスト | - |
| スタイルシート | - |
| フォーマット/フィルター | - |
| ライブラリーからのピクチャー | - |
| ヘルプtips | - |
| **---- エクスプローラーと結果ウィンドウ ----** | |
| プロジェクトフォーム | フォームメソッド |
| テーブルフォーム | フォームメソッド |
| プロジェクトメソッド | - |
| フォルダー/サブフォルダー | - |
| テーブル | フィールド, トリガー |
| **---- フォームエディター ----** | |
| すべてのフォームオブジェクト (ボタン, 変数等)。フォームを移動する際、含まれるオブジェクトもすべて移動されます。 | オブジェクトメソッド |

### 移動の動作

オブジェクトは標準のドラッグ＆ドロップやコピー/ペーストで移動できます。

ドラッグ＆ドロップを使用して2つのデータベース間でオブジェクトを移動するには、4Dアプリケーションを複製する必要があります。

データベース間の移動の場合、移動されるオブジェクトは、移動元と同じ環境 (ツールボックスやエクスプローラー等)、またはアプリケーションの他のエリアにペーストあるいはドロップできます。4Dは可能な限りコンテキストに従い適切な処理を行います。例えばフォームをメソッドエディターウィンドウにドロップできます。この場合、メソッドにフォーム名が挿入されます。

移動の間、同じタイプで同じ名前のオブジェクトが移動先のデータベースに存在すると、デフォルトで既存のオブジェクトが移動されたオブジェクトで上書きされます。この場合移動ダイアログボックスが表示され、オブジェクトが置換されようとしていることを示し、開発者はこのアクションを変更できます。

以下のメカニズムに留意してください:

- **表示とレベル**: 移動されたフォームオブジェクトはエディタ中の位置プロパティ (特に表示位置やフォームの異なるレベル) を保持します。
- **継承フォーム**: 継承フォームはソースフォームとともには移動されません。しかし参照は保持されます。さらに継承フォームは依存オブジェクトとして扱われ、移動が実行されたとき、他の (既存の) フォームを継承フォームとして使用できます (後述参照)。
- **アクセス権**: 移動されるフォームやプロジェクトメソッドは元のアクセス権を保持しません。自動でデフォルト値 ("すべてのグループ") となります。
- **フォルダー**: エクスプローラーのホームページからフォルダーを移動すると、フォルダーおよび底に含まれるすべての内容 (テーブル、フォーム、プロジェクトメソッド) が処理の対象となります。これは大量のデータ処理となる可能性があります。このタイプの移動の場合、この動作を示すために警告ダイアログボックスが表示されます。

**注:** エクスプローラーのゴミ箱ページからオブジェクトをドラッグすることはできません。

### 依存オブジェクト

- フォーム (テーブルおよびプロジェクト) はリスト、ピクチャーなど他のオブジェクトを参照することがあります。これらのオブジェクトは依存オブジェクトと呼ばれます。特定のケースで、すべての依存オブジェクトを移動する必要があるかもしれませんし、他の場合では、そのうちのいくつかあるいはまったくオブジェクトを移動しない場合もあるかもしれません。4Dではデータベース設定の移動ページや移動ダイアログボックスを使用して、依存オブジェクトの移動をコントロールできます。
  移動設定は依存オブジェクトの移動時に適用される原則を定義します。オブジェクトごとに様々なデフォルトオプションを指定できます。詳細は移動プロパティを参照してください。
- 4D v13より、メソッドで参照されているテーブルは、メソッドとともにデフォルトで移動されるようになりました。結果ウィンドウやエクスプローラーからメソッドをドラッグする際に**Shift**キーを押すことで、この動作を無効にすることができます。

## 移動ダイアログボックス

2つの4Dデータベース間やデータベースとオブジェクトライブラリー間で一連のオブジェクトを (ドラッグ＆ドロップやコピー-ペーストで) 移動するときに、移動されるすべてのオブジェクトと、各オブジェクトに割り当てられた移動先データベースで行われるアクションをリストするダイアログボックスが表示されることがあります。
このダイアログボックスは"移動ダイアログボックス"と呼ばれ、以下のいずれかのケースに当てはまる場合に表示されます:

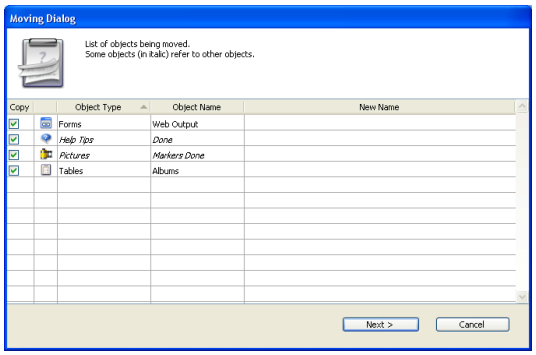
- 移動設定で"常に"表示が選択されている (WEB SET HOME PAGE参照)。
- 移動されるオブジェクトの少なくとも一つが、移動先データベースのオブジェクトと名前の衝突を起こしている。
- 移動される依存オブジェクトのすくなくとも一つに**他のオブジェクトを使用**デフォルトアクションが選択されている。

オブジェクトの移動が全く衝突を起こさない場合、移動ダイアログボックスは表示されず、オブジェクトは直接コピーされます。

このダイアログボックスではコンテキストに基づき、移動設定を確認して変更することができます。ダイアログボックスには2つのページ、**メインページ**と**詳細ページ**が含まれます。これらのページ間は**次へ>**と**<戻る** ボタンを使用して行き来できます。

### メインページ

メインページには移動されるオブジェクトがリスト表示されます:




名前の衝突があるオブジェクトは**太字**で、依存オブジェクトは*イタリック*で表示されます。列には各オブジェクトのタイプや名前、そして必要に応じて"新しい"名前 (移動先データベースでのデフォルトの名前) が示されます。詳細ページでこのデフォルトの名前を変更できます (次節参照)。
"コピー"列のチェックボックスは、オブジェクトが移動先データベースにコピーされるかどうかを示します。このオプションは設定可能です。このチェックを外すことで、特定のオブジェクトに関わる衝突の問題を素早く回避することができます。衝突が依存オブジェクトに関連する場合、その参照 (名前) は移動先オブジェクト内で保持されることに留意してください。**Alt+click** (Windows) または **Option+click** (Mac OS) を行うと、すべてのチェックボックスのチェック/チェック解除を切り替えることができます。

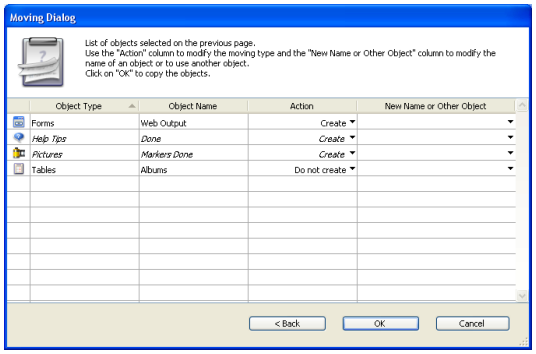
デフォルトの移動設定で問題ない場合、直接**OK**をクリックしてオブジェクトの移動を実行できます。

**注:** いずれかのオブジェクトに**他のオブジェクトを使用**アクションが割り当てられている場合、まず移動先データベース内での代わりのオブジェクトを指定しなければ (またはこのオブジェクトをコピー対象から外す)、コピー処理を開始することはできません。

特定のアクションを変更したい場合、**次へ>** ボタンをクリックして詳細ページを表示します。オブジェクトの移動をキャンセルするには**キャンセル**をクリックします。

### 詳細ページ

詳細ページにはコピーされるオブジェクト (メインページでチェックされたままになっているオブジェクト) がリストされ、移動設定を変更することができます:




"アクション"列のポップアップを使用してそのオブジェクトに対して実行されるアクションを変更できます。メニューに表示されるアクションは選択されたオブジェクトのタイプにより異なります。
一回の処理で複数の行に割り当てられたアクションを変更できます。これを行うには変更する行を選択し、そのいずれかの"アクション"列で新しいアクションを選択します。変更は適切である限り選択されたすべての行に適用されます。いずれかの行のアクションに互換性がない場合、その行は変更されず、警告ダイアログボックスが表示されます。

"新しい名前または他のオブジェクト"列には、コピー先データベースにオブジェクトがコピーされるときに割り当てられる名前が表示されます。この名前は変更できますが、移動先データベースに既に存在する名前を使用しないように注意しなければなりません。そうでなければ新しい名前の衝突が発生してしまいます。依存オブジェクトの場合、この列を使用して移動先データベースの他のオブジェクトを指定することもできます (**他のオブジェクトを使用**アクションが選択されている場合)。例えばテーブルフォームを移動する場合、テーブルを作成するのではなく、属するテーブルを指定するために、移動先データベースのテーブルを指定することができます。

#### 選択可能なアクション

以下のアクションを選択することができます:

- **作成しない**: オブジェクトはコピーされません。依存オブジェクトの場合、データベース中に同じ名前のオブジェクトが既に存在する場合、その参照 (名前) は保持されます (この場合それがメインオブジェクトによって使用されます)。同じ名前のオブジェクトを利用できない場合、参照は削除されます。
- **置換**: このオプションは同じタイプで同じ名前のオブジェクトがデータベース中に既に存在する場合に提供されます。この場合、移動先データベース中のオブジェクトは移動元データベースのオブジェクトで置き換えられます。
- **作成**: 依存オブジェクトはプロパティとともに移動先データベースにコピーされます (オプションは名前の衝突がない場合に提供されます)。
- **作成して名称変更**: このオプションは同じタイプで同じ名前のオブジェクトがデータベース中に既に存在する場合に提供されます。デフォルトで、オブジェクトは名前に番号を付加することで名称変更されます。この場合、"新しい名前または他のオブジェクト"列を使用して移動するオブジェクトの名称を変更できます。オブジェクト参照は移動先データベース内で更新されます。
- **他のオブジェクトを使用**: このオプションは依存オブジェクトでのみ利用できます。このオプションは、参照として移動先データベースに既に存在する他のオブジェクトを利用可能にします。この場合、"新しい名前または他のオブジェクト"列には使用できる他のオブジェクトのリストが表示されます。
- **同じ名前のテーブルを使用**: このオプションはデータベースに同じ名前のテーブルが既に存在する場合に提供されます。この場合、"新しい名前または他のオブジェクト"列には依存テーブルの代わりに使用できるテーブルのリストが表示されます。

依存オブジェクト自身が他のオブジェクトを参照している場合、設定に基づいてリストが更新されます。

移動設定に問題がない場合、**OK**をクリックしてオブジェクトの移動を実行できます。**<戻る**をクリックするとメインページに戻ります。オブジェクトの移動をキャンセルするには**キャンセル**をクリックします。

## 移動プロパティ

データベース設定の移動ページを使用して、デザインモードでオブジェクトを移動した場合のデフォルト動作を設定できます。4Dはこのデータベース (移動先データベース) にドロップ/ペーストされたオブジェクトに対しこれらの設定を適用します。

**依存するオブジェクトをコピーする際のデフォルトアクション**

これらのオプションは依存オブジェクト、言い換えれば移動されるフォーム (CHANGE LICENSES参照) にリンクされたオブジェクトの移動を設定します。

これらのデフォルトアクションは、オブジェクトの移動時にコンフリクトが発生せず、**名前の衝突時のみ**が移動ダイアログを表示で選択されている場合 (次節参照)、自動で適用されます。そうでない場合、移動ダイアログが表示され、その中でデフォルトの選択肢として使用されます。

4Dは各オブジェクトごとに**無視**、**作成（必要なら名前を変更）**、**作成（必要なら置換）**、そして**他のオブジェクトを使用**オプションを提供します。移動ダイアログボックスが表示される場合、オブジェクトごとにより詳細なオプションが提供されることに留意してください。

- **無視**: この場合依存オブジェクトは移動先データベースにコピーされません。移動ダイアログにおいてデフォルトで**作成しない**アクションが選択されます。
- **作成（必要なら名前を変更）**: この場合依存オブジェクトは常に移動先データベースにコピーされます。オブジェクトが移動先データベースに存在しない場合、移動ダイアログにおいてデフォルトで**作成**アクションが選択されます。
  移動先オブジェクトにすでにその名前を持つオブジェクトが存在する場合、コピーされるオブジェクトの名前には接尾辞"_X"が追加されます。この場合、移動ダイアログにおいてデフォルトで**名称変更**アクションが選択されます。
- **作成（必要なら置換）**: この場合依存オブジェクトは常に移動先データベースにコピーされます。オブジェクトが移動先データベースに存在しない場合、移動ダイアログにおいてデフォルトで**作成**アクションが選択されます。
  移動先オブジェクトにすでにその名前を持つオブジェクトが存在する場合、既存のオブジェクトはコピーされるオブジェクトで上書きされます。この場合、移動ダイアログにおいてデフォルトで**置換**アクションが選択されます。
- **他のオブジェクトを使用**: このオプションを選択すると、名前の衝突時のみが選択されていても、システマチックに移動ダイアログが表示されます。オブジェクトを移動する際、コピーされる依存オブジェクトの代わりに使用する移動先データベースのオブジェクトを選択しなければなりません。

**注:** これらのオプションは依存オブジェクトのみを対象とします。移動されるオブジェクト自身のデフォルトアクションは**作成（必要なら名前を変更）**です。

- **移動ダイアログを表示**: このメニューでは移動ダイアログボックスの表示を設定します。
  **常に**オプションが選択されていると、オブジェクトが移動されるたびに移動ダイアログボックスが表示されます。これにより移動操作をより詳細に設定できます。**名前の衝突時のみ**オプションを選択すると、移動される (依存またはメインの) オブジェクトが移動先データベースのオブジェクトと名前の衝突を起こしている場合のみ、ダイアログが表示されます。

# データベースストラクチャーの作成


データベースの基本

ストラクチャーエディター

インスペクターパレット

テーブルの作成と変更

テーブルプロパティ

プライマリーキー管理

フィールドの作成と変更

4Dのフィールドの型

フィールドプロパティ

データをデータファイル外に保存

テーブルとフィールドの命名規則

インデックスの作成と変更

リレーションの作成と変更

リレーションのタイプ

リレーションプロパティ

ストラクチャー定義の書き出しと読み込み

# データベースの基本

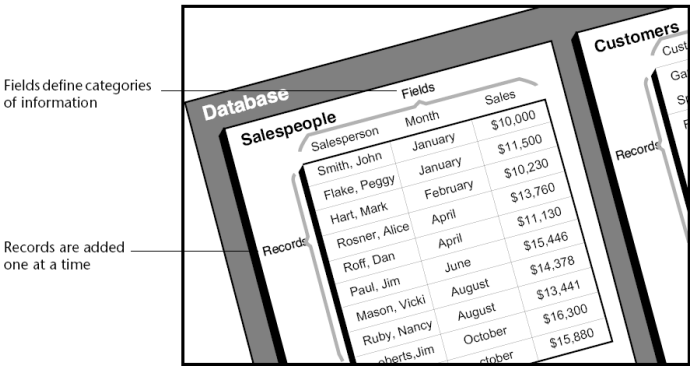
データベースは系統的な情報の集合体です。そのため効率よく使用することができます。電話帳はデータベースの良い例です。他にも辞書、カレンダー、レシピ集などがあります。

データベース上の情報は、レコード形式で構成されます。データベースの各レコードには、人物や物に関する情報のすべてが格納されています。例えば、電話帳の各レコードには、ある人物の名前、住所、電話番号が格納されています。

各レコードにはフィールドがあります。フィールドは特定の情報を格納するために使用します。例えば、電話帳のデータベースでは、1つのフィールドにある人物の名前を格納し、2つ目のフィールドにその人物の住所を格納し、3つ目のフィールドにその人物の電話番号を格納します。すべてのレコードはそれぞれこれらのフィールドを含み、またすべてのレコードはこれらのフィールドに情報を持つことができます。

通常フィールド名は、フィールドに格納される情報を識別します。通常フィールド名はName、Address、あるいはPhoneNumberのようになります。各フィールドはフィールドタイプを持っており、数値、日付、文字など、フィールドに入力できる情報の種類を特定します。各フィールドには特定のタイプのデータが含まれるため、フィールド上の情報を使用して計算や操作を行うことができます。例えば、2つのフィールドの数値を加算することができます。あるフィールドの日付を別のフィールドの日付と比較できます。住所ラベルの最初の行を作成するために、姓 (あるフィールドに保存されている) を名 (別のフィールドに保存されている) を結合できます。

すべてのレコードが集まり、1つのテーブルを構成します。データベースには多くのテーブルを含むことができます。以下の挿絵は、どのようにこれらのコンセプトが関連しているかを表しています。



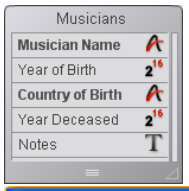

情報を役立つものにするため、4Dはレコードを再編成し、情報を基に計算を行なうことができます。例えば4Dはフィールドの総計を計算し、レポートに表示することができます。各営業マンの売上合計を計算し、売上高を比較するグラフを表示することができます。

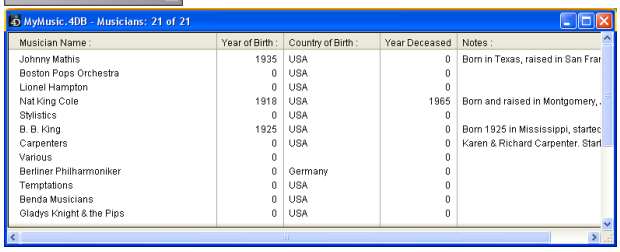
## テーブルとフィールド

4Dは1つのデータベースにつき1から32,767までのテーブルを作成できます。これはあなたのニーズに適合したストラクチャーを作成できることを意味します。

### シングルテーブルのストラクチャー

いくつかのデータベースは、1つのテーブルのみを使用します。従業員、企業、在庫など1つのカテゴリー情報に対して1つのテーブルを使用します。テーブルには必要な数のフィールドを持つことができます (32,767まで) 。




上の図では、同じタイプのデータがすべての従業員のレコードに対して必要です。データベースは、保存されている従業員の人数に応じて拡大します。

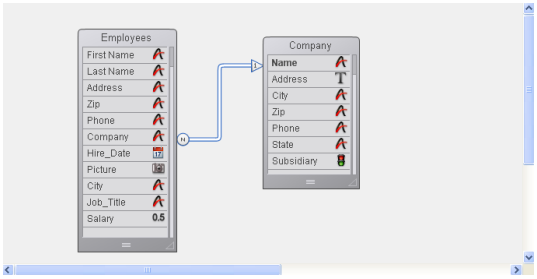
### 複数テーブルのストラクチャー

データベースは時に2つ以上のテーブルを使用すると、より効率よくデータを格納し、データへアクセスできるようになります。異なるタイプの情報は別のテーブルに格納するというルールを覚えておくとよいでしょう。

従業員と企業の2つの情報を追跡するデータベースがよい例です。従業員と企業のレコードは別のテーブルに格納されます。企業の所在地が変更された場合、企業のレコードのみを変更する必要があります。その企業の全従業員に対して新しい所在地を入力する必要はありません。

シングルテーブルでは、各個別のレコードに所在地を入力する必要があります。2つのテーブルでは、一度だけ情報を入力する必要があります。企業名が従業員のレコードに入力されると、4Dは企業のレコードを検索し、自動で正しい所在地を表示します。

以下の図は、2つのテーブルがリレートしている複数テーブルのデータベーススストラクチャーを表しています。[Employees]Companyフィールドと[Company]Nameフィールドの間に描かれた矢印は、リレーションを示しています:




各従業員のデータは、[Employees] テーブルに保存されています。各企業のデータは、別途[Company] テーブル1に保存されています。

4Dはリレーショナルデータベースアプリケーションと呼ばれています。それは、複数のテーブルを用い、さまざまな方法でそれらのテーブルを関連付けるからです。例えば[Company] テーブルを検索する[Employees] テーブルのレポートを作成し、各従業員の会社に関する情報を自動的に表示し印刷できます。テーブル間のリレーションにより、各テーブルの情報がレポートで使用できるようになります。

また、リレートされたテーブルにデータを直接入力できます。例えば請求書データベースで、請求書画面内から[LineItems] テーブルに情報を書き込むことができます。更に4Dランゲージを使用してリレートされたテーブルにデータを書き込むこともできます。

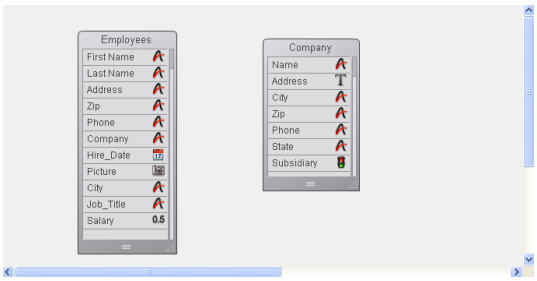
時には、テーブルが直接リレートされていない複数テーブルのストラクチャーを必要とすることがあります。例えば連絡先リストや経費テーブルなどのさまざまな情報を1つのデータベースに格納させると便利な場合があります。

4Dでは各データベース上に32,767個までのテーブルを作成することができます。1つのテーブルには32,767個までのフィールドを作成することができます。複数テーブルを使用すると、事実上ほぼあらゆるタイプのデータベーススストラクチャーを作成できます。

## テーブルをリレートする

通常複数のテーブルが情報を共有するストラクチャーを作成する必要があります。例えば従業員と彼らが働く会社について追跡するデータベースを作成するケースについて考えてみましょう。

データベーススストラクチャーには以下の図のように、従業員の情報を格納する[Employees]テーブルと会社の情報を格納する[Companies]テーブルが含まれます。

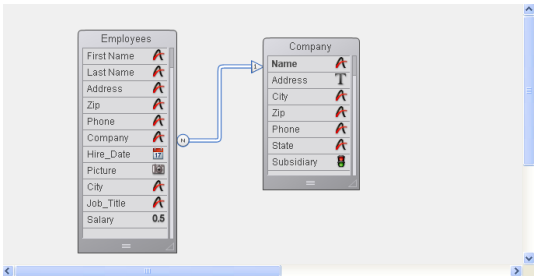



各々分割されたテーブルに格納された情報だけでは、情報追跡のニーズを満たすことはできません。[Employees]テーブルのレコードを表示する際、その従業員が働く会社の情報も表示する必要があります。また[Companies]テーブルを表示する際には、その会社で働く全従業員の情報を表示する必要があります。
このような形で2つのテーブル間で情報を共有するために、テーブルをお互いにリレートさせることができます。言い換えると、リレーションは各テーブルのデータ間に設定することができます。4Dではテーブルリレーションを使用して、あるテーブルに格納されたデータに他のテーブルのデータからアクセスできます。リレーションに基づき情報を共有するテーブルをリレートしたテーブルと呼びます。

テーブルをリレートすると以下のことが可能となります:

- データを効率的に格納する。
- 一か所のデータを更新して、そのデータを使用するすべての場所に変更を反映させる。
- リレートする情報を表示する。
- 他のテーブルのデータに基づき他のテーブルに対し検索や並び替えを実行する。
- リレートしたテーブルのレコードを作成、更新、削除する。

下の図ではストラクチャーエディターで[Employees]テーブルと[Companies]テーブル間にひかれたリレーションを示しています:




[Employees]テーブルには従業員ごとに1つのレコードがあります。[Companies]テーブルには会社ごとに1つのレコードがあります。2つのテーブル間のリレーションを使用すれば両サイドのテーブルから情報にアクセス、入力、変更、および削除を行えます。例えば:

- 従業員レコードが表示されているとき、対応する会社の情報 (所在地、郵便番号、電話番号など) を表示して、更新することができます。
- 新しい従業員を追加するとき、従業員レコードを適切な会社レコードにリンクできます (会社が既に存在している場合)。会社レコードがまだデータベースに登録されていないときには、従業員レコードを作成中に会社レコードも作成することができます。詳細はリレーションプロパティを参照してください。
- 会社レコードごとに、その会社に所属する従業員を表示し、情報 (名前、役職、電話番号等) を更新できます。会社レコードから従業員レコードを追加することもできます。

### リレートしたフィールド

リレートしたフィールドとは、2つのテーブルをリレーションによって結び付けるフィールドです。リレートしたフィールドを使用することで、リレートしたテーブルの情報を表示できます。
テー ブルをリレートする基本的な目的は、あるテーブルのカレントレコードに対応して、他のテーブルのどのレコードをカレントにすればよいか、その判定方法を 4Dに指示することにあります。リレートしたテーブルは、2つのリレートしたフィールドのデータを使用して、対応するレコードを識別します。
先の 例題では、会社名が[Employees]テーブルと[Companies]テーブル両方に格納されていました。[Employees]テーブルの Companyフィールドと[Companies]テーブルのNameフィールドが2つのテーブルをリレートしています。[Companies]テーブル のNameフィールドは[Companies]テーブルの主キーです。このフィールドの値を使用して、それぞれの会社のレコードをユニークに識別します。 主キーにはインデックスおよび重複不可属性を割り当てるべきです。[Employees]テーブルのCompanyフィールドは外部キーフィールドです。 外部キーフィールドの各値は、リレートしたテーブルの主キーフィールドの1つの値と正確に対応します。リレーションを作成する際、主キーにインデックス属 性が割り当てられていないと、4Dが自動で割り当てます。

外部キーフィールドのそれぞれの値は、リレートした他のテーブルの主キーフィール ドの1つの値と等しくなります。この例題では、[Employees]の外部キーフィールドの値は、[Companies]の主キーフィールドの1つの値 と完全に一致します。外部キーの値は重複不可ではありません (同じ会社の中で複数の従業員が働くことができるからです)。

4D v14以降、データベースのそれぞれのテーブルにおいて、プライマリーキーが明示的に定義されなければなりません。プライマリーキーフィールドの値は通常(4Dが生成する一連番号または開発者が記述したコードが生成するキーを割り当てることで)データベースによって自動的に割り当てられます。このような方法はキーフィールドがユニークであることを保証します。例えば [Companies]テーブルの主キーが会社名ではなく一連番号であれば、ユーザーは同じ会社名であっても所在地が異なる複数の会社を入力することが可 能となります。また会社名が変更になった場合も、2つのテーブル間のリレーションを壊すことなく、データベースに変更を行うことができます。

ユーザーが主キーフィールドに値を入力することができる設計の場合、フィールドプロパティ にて**重複不可と更新不可**属性の両方をチェックし、最初の入力で重複する値がないことを検証し、後でこの値が変更されないように設定すべきです。更新不可属性を使用しない場合、主キーフィール ドの値が変更されたときに、リレートしたテーブル中でレコードが迷子にならないよう、別途手当てをする必要があります。

リレーションが設 定されると、あるテーブルで作業をしている間に、リレートしたテーブルの値を読み書きすることができるようになります。例えば従業員レコードに会社名を入 力すると、4Dは[Companies]テーブル中でそのレコードを検索し、その従業員レコードに会社の所在地と電話番号を表示します。会社レコードを表 示しているとき、4Dはその会社で働いている従業員を[Employees]テーブルから検索し、会社レコードに表示します。

これらのリ レーションは自動で実行できます (プログラムは必要ありません)。あるいはマニュアルリレーションを選択することもできます。後者の場合、リレートしたレコードのロードやアンロードはメ ソッドを使用して行い、リレートしたレコードの作成や変更、削除を制御できます。マニュアルリレーションは、2つ以上のテーブルが1つのテーブルにリレー トされているなど複雑なストラクチャーで利用されます。またリレートしたレコードのロードやアンロードを制御したい場合にも使用されます。自動またはマ ニュアルリレーションどちらを使用するかは、リレーションのプロパティで設定できます (リレーションプロパティ参照)。

# ストラクチャーエディター

ストラクチャーエディターには**デザイン**メニューの**データベースストラクチャー**コマンドを選択するか、4Dツールバーの ボタンをクリックしてアクセスできます。

ストラクチャーエディターを使用してデータベースストラクチャー (テーブルとリレーション) を管理できます。このエディターを使用してテーブル、テーブルプロパティ、フィールド、フィールドプロパティ、テーブルリレーションを管理できます。

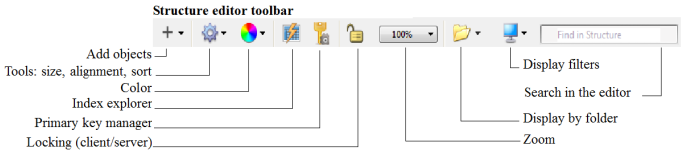
ストラクチャーエディターはデータベースストラクチャーをグラフィカルに表示し、データベースデザイン操作に使用するツールバーやコンテキストメニューを提供します。

ストラクチャーエディター中で、それぞれのテーブルはテーブルイメージとして表現されます。テーブルにはフィールドとその型がアイコンで表示されます。マウスを重ねると、情報バーにテーブルやフィールドの情報が表示されます。

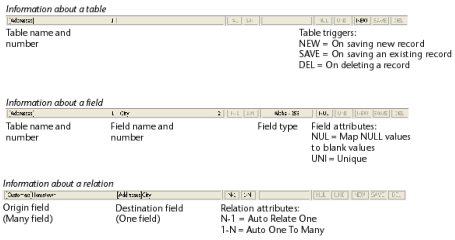
フローティングパレット上に表示されるインスペクターを使用して、ストラクチャーオブジェクトやストラクチャーエディター自身のプロパティを表示したり変更したりできます。このウィンドウについてはインスペクターパレットで説明します。

### ツールバーと情報バー

ストラクチャーエディターにはオブジェクト追加などの機能、およびナビゲーションやオプションを表示するツールバーがあります:




エディターウィンドウの下部にはマウスが重なっているオブジェクト (テーブル、フィールド、リレーション) のエリアに対応する情報を表示する情報バーがあります:




### オブジェクトを選択する

ストラクチャーエディター中のオブジェクトのイメージを操作するには、まずそれを選択します。そしてそのプロパティを定義したり、移動したり、サイズ変更したり、削除したりできます。同時に同じタイプのオブジェクトを複数選択して、例えばインスペクターパレットで共通のプロパティを表示し、更新することができます。異なるテーブルのフィールドを同時に選択することもできます。

テーブルを選択するには:

1. テーブルイメージをクリックする、
   または
   **[Tab]**または**Shift+[Tab]**を押して、ストラクチャーエディター中のテーブルを移動しながら選択します。
   選択されたテーブルの外周部は青くなります。以降の操作は選択したテーブルに対して効果を及ぼします。

フィールドまたはリレーションを選択するには:

1. フィールドまたはリレーション上でクリックする、
   または
   テーブルが選択されているとき、**[upper arrow]**または**[lower arrow]**キーを押してテーブル中のフィールドを移動して選択する、
   または
   テーブルが選択されているとき、**[Home]**または**[End]**キーを押してテーブルの先頭や最後のフィールドを選択します。

同じタイプのオブジェクト (テーブル、フィールド、リレーション) を複数選択するには:

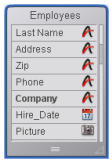
1. 連続したオブジェクトを選択するには**Shift+click**を使用します。
   または
   連続しないオブジェクトを選択するには**Ctrl+click** (Windows) または**Command+click** (Mac OS) を使用します。
   または
   空のエリアをクリックし、選択するオブジェクトの周りに四角を描画します (テーブルの選択のみ)。
   または
   **編集**メニューまたはエディターのコンテキストメニューから**すべてを選択**コマンドを選択します (テーブルの選択のみ)。

### テーブルイメージを操作する

必要に応じて、ストラクチャーエディター中でテーブルイメージのサイズを変更したり移動したりできます。

#### フィールドリストのスクロール

テーブルにフィールドを追加するときや、テーブルのサイズを小さくしたときなど、テーブルイメージが表示できるよりも多くのフィールドがテーブルに含まれることがあります。このような場合、4Dは自動でスクロールバーを追加し、フィールドリストをスクロールして表示できるようにします:




#### テーブルイメージのサイズ変更

テーブルイメージのサイズを変更して、より多くのフィールドを表示したり、画面を再構成したりできます。サイズ変更には複数の方法があります:

- **手動によるサイズ変更**: テーブルイメージの下部または右下の角をクリックしドラッグします。

- **最適サイズ**: これは選択されたテーブルを自動でサイズ変更し、含まれるフィールド全体がぴったりひょ応じされるようにします(空行は表示されません)。
  **最適サイズ**コマンドを適用するには以下の手順を行います:
  - エディタのツールバーボタンに割り当てられたメニューから**最適サイズ**コマンドを選択する (テーブルを選択していない場合このコマンドは使用できません)。

    
  - エディターのコンテキストメニューから**最適サイズ**コマンドを選択します (テーブル上でクリック)。
  - テーブルの名前部分を**Shift+ダブルクリック**します。この操作を繰り返すと、以下の動作が適用されます: 元のサイズ -> 最適サイズ - > 折りたたむ
- **折りたたむ**: この機能はテーブルイメージを折りたたみ、テーブル名のみが表示されるようにします。これは大きなストラクチャーで便利です。

**折りたたむ**コマンドを適用するには、以下を行います:

- エディターのコンテキストメニューから**折りたたむ**コマンドを選択します (テーブルのタイトルエリア上でクリック)。
- テーブルの名前部分を**Shift+ダブルクリック**します。この操作を繰り返すと、以下の動作が適用されます: 元のサイズ -> 最適サイズ - > 折りたたむ

**注:**

- テーブル名エリアでダブルクリックを行うとインスペクターが表示されます。
- **Alt** (Windows) または **Option** (Mac OS) + ダブルクリックでテーブルのトリガーメソッドがメソッドエディタ中で開かれます。

**4D Server:** 4D Server環境でテーブルイメージをサイズ変更すると、すべてのユーザーのデザインモードでテーブルイメージサイズが変更されます。

#### テーブルイメージの移動

機能に基づいてテーブルをグループ化したり、ストラクチャーエディターウィンドウを再編成したりするために、テーブルイメージをエディター内で移動することができます。ストラクチャーの可読性を向上させるためにテーブルを揃えることもできます。リレーションは新しいテーブル位置に基づいて再描画されます。

手動でテーブルイメージを移動するには:

1. マウスカーソルでテーブル名エリアをクリックしドラッグします。
   テーブル名バーのみをドラッグします。テーブルイメージの他の部分をドラッグすると、異なる効果を起こします。例えば新しいテーブルリレーションが作成されたりテーブルイメージのサイズが変更されたりします。

**4D Server:** 4D Server環境でテーブルイメージを移動すると、すべてのユーザーのデザインモードでテーブルは新しい位置に表示されます。

複数のテーブルの位置を揃えるには:

1. 揃えるテーブルを選択し、ツールメニューから対応するメニューを選択します:

   
   これらのコマンドは2つ以上のテーブルが選択されている場合に有効となります。

### スクロール

巨大なストラクチャーにおいては特別なスクロール機能がナビゲーションを容易にします:

- マウスホイールを使用するとエディターウィンドウが縦にスクロールされます。カーソルがテーブル上にあるときはテーブルのフィールドがスクロールされます。
- マウスホイールスクロール時に**Shift**を押すと、横にスクロールされます。
- **Shift**キーを押して空のエリアをクリック＆ドラッグするとハンドツールが使用可能になり、ウィンドウの内容全体をスクロールすることができます。

### ズーム

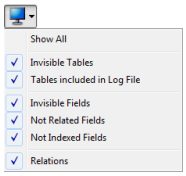
ストラクチャーツールバーのズーム機能を使用して、データベーススストラクチャーの表示倍率を変更できます。データベースを開く際のデフォルトは100%です。セレクションがフォーム内にある場合にはそこへズームされます。

ズーム設定はユーザーごとに異なります。設定はウィンドウを閉じる際に保存されます。

**注:** 環境設定を使用してズーム時のストラクチャーエディターの描画品質を設定できます。詳細はストラクチャーページを参照してください。

### オブジェクトタイプの表示

ストラクチャーエディターのツールバーの表示ボタンには、タイプに基づいてストラクチャーオブジェクトの表示と非表示を切り替えるメニューが割り当てられています:



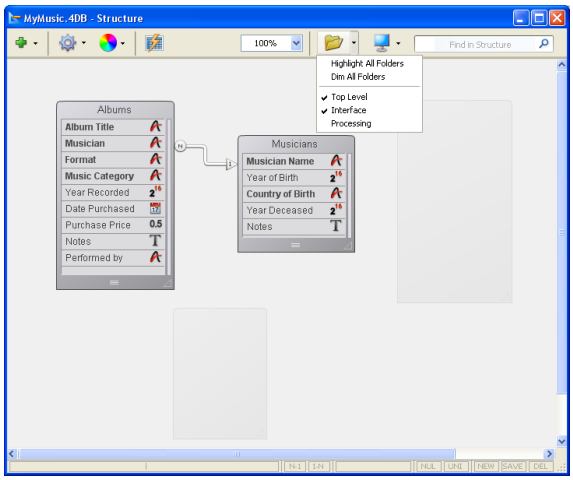

デフォルトですべてのオブジェクトが表示されます。この機能を使用してさまざまな表現 (もっとも単純なものから完全な表示まで) を得ることができます。そして必要な情報のレベルに基づくストラクチャーの検証ツールを提供します。選択はすべてのテーブルとフィールドに適用されます。設定はユーザーごとに異なり、ウィンドウを閉じる際に保存されます。特定のオブジェクトを非表示にしてもテーブルの位置は変わりません。

- 複数のフィールドタイプがチェックされていると、論理OR演算を使用して表示するオブジェクトが決定されます。例えば**非表示フィールドとインデックスされていないフィールド**オプションが選択されている場合、(表示・非表示属性に関わらず) すべてのインデックスがないフィールドと、(インデックス属性に関わらず) すべての非表示フィールドが表示されます。
- テーブルはフィールドに対して優先されます。テーブルが表示されない場合、フィールドも表示されません。
- 表示されないタイプのオブジェクト (テーブル、フィールド、リレーション) を追加すると、オブジェクトはエディターに表示されます。それを非表示にするには、**表示**ボタンに割り当てられた対応するオプションを改めて設定する必要があります。

### フォルダーごとにテーブルをハイライト/薄暗くする

エクスプローラーウィンドウのホームページページで定義されたフォルダーに基づき、ストラクチャーエディター内でテーブルグループを非表示にすることができます。ストラクチャーエディターのツールバーにあるフォルダーボタン を使用してテーブルの表示を設定します。

ボタンに割り当てあられたメニューには表示を管理するコマンドと、データベースに定義されたフォルダーのリストが表示されます。表示されるフォルダーの隣にはチェックが付けられます。このメニューでフォルダーを選択/選択解除することで、表示を変更できます。
**すべてのフォルダーをハイライト/すべてのフォルダーを薄暗く表示**コマンドを使用してデータベースのフォルダー中のテーブルをすべて表示したり隠したりできます。
ボタンをクリックするたびにテーブルの表示が切り替えられます。
テーブルが薄暗く表示されているとき、デフォルトでストラクチャーウィンドウにはその外周部だけが表示されます。隠されていないテーブルのみが全体として表示されます:




**注:** ストラクチャーエディター中で (フォルダーごとに) 薄暗く表示するテーブルのアピアランスを、環境設定のストラクチャーページにあるオプションで切り替えることができます。この環境設定を変更した場合には設定を有効にするために、ストラクチャーエディターウィンドウを一度閉じて開きなおす必要があります。

### エディターウィンドウのカスタマイズ

ストラクチャーエディターの特定のプロパティはインスペクターを使用して変更することができます。

ストラクチャーエディターウィンドウのプロパティを表示するには:

- ウィンドウの空のエリアをダブルクリックします。またはインスペクターウィンドウが既に表示されている場合は空のエリアをクリックします。
- ウィンドウの空のエリアで右クリックをし、コンテキストメニューから**ストラクチャープロパティ**コマンドを選択します。

インスペクターパレットにストラクチャープロパティが表示されます:

以下のプロパティを設定できます:

- **バックグランドピクチャー:** 使用するバックグランドピクチャーとその表示フォーマットを変更できます。
  ピクチャーを変更するには、**開く**ボタンをクリックするかプレビューエリアをクリックしてコンテキストメニューから**開く**を選択します。そして表示するピクチャーを選択します。どのピクチャーフォーマットも使用できます。選択されたピクチャーをすぐにプレビューエリアとエディターウィンドウに表示されます。
  ピクチャーの表示フォーマットを変更するには、**ピクチャーフォーマット**メニューから値を選択します。提供される選択肢は4D標準のピクチャー表示フォーマットです。
  カスタムピクチャーを削除するには、**クリア**ボタンをクリックするか、プレビューエリアをクリックしてコンテキストメニューから**クリア**を選択します。
- **塗りカラー:** エディターウィンドウのバックグランドに使用されるカラーを変更できます。これを行うにはカラー選択エリアをクリックして選択メニューからカラーを選択します。
- **デフォルトフォント**: テーブルやフィールド名に使用されるデフォルトのフォントを変更するには、このメニューから使用するフォントを選択します。サイズメニューを使用してデフォルトのフォントサイズを変更することもできます。

**ストラクチャーを検索**

4Dではストラクチャーエディターウィンドウ内で検索を行えます。検索は以下の要素に対して行われます:

- テーブル名やフィールド名
- テーブル番号

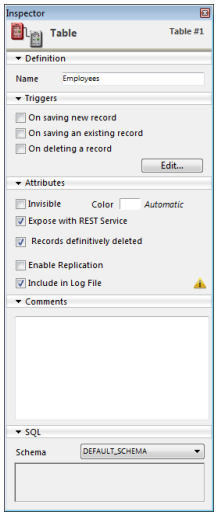
検索を行うには、ストラクチャーを検索エリアに検索する文字列やテーブル番号を入力します。エリアに値を入力するとその下にオプションバーが表示され、検索のスコープとタイプを指定できます:

- **検索**メニューは検索のスコープを設定します (**テーブル&フィールド**または**テーブル**のみ)。
- **オプション**メニューは検索のタイプを設定します:
  - **含む** (デフォルト): "le"を検索すると"Table"、"Letter"、"Elements"等が見つかります。
  - **前方一致**: "pa"を検索すると"paper"や"paid"が見つかりますが、"repair"は見つかりません。
  - **番号**: "2"を検索するとテーブル番号2、12、20、21が見つかります。

検索は値を入力するに従いリアルタイムで実行されます。検索されたテーブルやフィールドは強調表示されます。何も見つからないと、検索条件エリアが赤に変わります。検索モードを抜けるには、オプションバー (Windows) または検索条件エリア (Mac OS) の ボタンをクリックするか、エリア内の文字をすべて削除します。

## インスペクターパレット

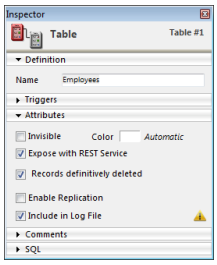
ストラクチャーエディターオブジェクト (テーブル、フィールド、リレーション) のプロパティはインスペクターパレットに表示し編集することができます:




このパレットはオブジェクト上でダブルクリックすると表示されます。表示内容は選択されたオブジェクトに基づきダイナミックに更新されます。

ウィンドウの空のエリアをクリックすると、インスペクターにはストラクチャーエディターの一般的なプロパティが表示されます (ストラクチャーエディターの"エディターウィンドウのバックグランドのカスタマイズ"参照)。

パレットの異なるエリアは、パネルごとに展開・折りたたみができます。すぐとなりの三角形をクリックすることで対応する情報を表示、または非表示するができます:




以下のショートカットが使用することができます:

- 折りたたまれたパネルのタイトルバーを**Shift+クリック**すると、そのパネルを展開し、他を折りたたみます。
- 折りたたまれたパネルのタイトルバーを**Alt** (Windows) または**Option** (Mac OS) **+ クリック**すると、すべてのパネルを展開します。
- 展開されたパネルのタイトルバーを**Alt** (Windows) または**Option** (Mac OS) **+ クリック**すると、すべてのパネルを折りたたみます。

パレットの位置とパネルの展開/折りたたみ状態は保存されます。

# テーブルの作成と変更

いつでもテーブルを作成することができます。4Dは作成されたテーブルに[Table_1]と名前を付け、次にテーブルを作成する際にその番号が存在すれば番号をインクリメントします。テーブル名は変更することができます。

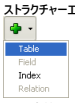
必要のないテーブルや間違って作成したテーブルは削除することができます。この削除は恒久的あるいは一時的にできます。恒久的にテーブルを削除すると、テーブルに格納されたデータも削除されます。テーブルを非表示にすることで、アプリケーションモード内のエディターにそのテーブルが表示されないようにすることもできます。

必要に応じてテーブル毎にトリガーや属性を設定できます。これらのプロパティーはテーブルプロパティで説明しています。

## テーブルの作成

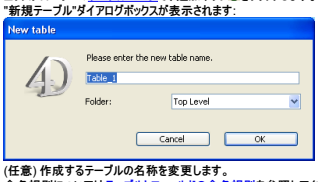
直接ストラクチャウィンドウで、あるいはダイアログボックス経由で、テーブルを作成できます。直接の作成は素早く行えますが、ダイアログボックスを 使用した作成ではテーブルの特定のパラメターを設定したり、必要であれば作成をキャンセルすることができます。またコピー/ペーストでテーブルを作成する こともできます。

直接テーブルを作成するには:

1. ストラクチャーエディターウィンドウの空のエリアで右クリックをし、コンテキストメニューから**テーブル追加...**を選択します。
   または
   ストラクチャーエディターツールバーのオブジェクト追加メニューから、テーブルを選択します:
   
   
   4Dは直接ストラクチャーエディター内にテーブルを作成します。その後、名前を変更したりフィールドを追加したりできます。

**注:** 特定のフォルダーにテーブルを入れたい場合、エクスプローラーのホームページを使用します。

"新規テーブル"ダイアログボックスを使用してテーブルを作成するには:

1. **ファイル**メニューの**新規 > テーブル**コマンドを選択します (または4Dツールバーの新規ボタンをクリックします)。
   または
   エクスプローラーのテーブルページで、追加ボタンをクリックします。
   "新規テーブル"ダイアログボックスが表示されます:
   
   
2. (任意) 作成するテーブルの名称を変更します。
   命名規則についてはテーブルとフィールドの命名規則を参照してください。
3. (任意) 新しいテーブルを配置するフォルダーを選択します。
   デフォルトでテーブルふぁフォルダー階層のトップレベルに配置されます。オブジェクトフォルダーに関する詳細はホームページを参照してください。
4. **OK**をクリックします。
   処理をキャンセルしたい場合は**キャンセル**ボタンをクリックします。
   4Dは新しいテーブルイメージを作成し、それがストラクチャーエディターウィンドウ内で選択されます。

**注:** 4D v14以降、データベース内で作成されたテーブルはどれもデフォルトでプライマリーキーフィールドを持ちます。この点についての詳細は、デフォルトのプライマリーキーフィールドを参照して下さい。

コピー/ペーストでテーブルを作成するには:

1. 複製するテーブルを選択し、標準の"コピー"コマンド (4Dの**編集**メニュー、コンテキストメニュー、または**Ctrl+C**/**Command+C**ショートカット) を実行します。
2. そして**編集**メニュー、コンテキストメニュー、または**Ctrl+V**/**Command+V**ショートカットで**ペースト**を実行します。
   テーブルはすべてのフィールドとともにペーストされます。デフォルトでテーブルの名称は"Copy(X)_of_*TableName*"となります。*TableName*は元のテーブル名で、Xはこのテーブルのコピー数です。

## テーブルの名称変更

テーブルはいつでも名称変更できます。古いテーブル名がメソッド中で使用されていた場合、4Dは自動でそれを新しい名称で置き換えます(メソッドがエディタで開かれていてはいけません)。メソッドが開かれている場合、手作業で変更を行う必要があります。データベース内でテーブル名はユニークでなければなりません。

テーブル名の変更はストラクチャーエディター、エクスプローラー、またはインスペクターパレットを使用して行うことができます。

テーブル名を直接変更する (ストラクチャーエディターまたはエクスプローラーのテーブルページ):

1. テーブル名を2回クリックします (テーブルが既に選択されている場合は1回クリック)。
   編集モードにスイッチします:

   
2. 新しい名称を入力し、テーブルの外側をクリックします。

ストラクチャーエディターのインスペクターパレットを使用して名称変更する:

1. テーブル名をダブルクリックする。
   または
   テーブルイメージ上で右クリックをし、コンテキストメニューから**テーブルプロパティ...**を選択します。
   または
   エクスプローラーの**テーブル**ページ上で、テーブル名をダブルクリックします。
   4D はインスペクターパレット内に選択されたテーブルの名前とプロパティを表示します。このパレットにはテーブル番号も表示されます。インスペクターペレット が既に開かれていて、他のテーブルや他のオブジェクトのプロパティを表示している場合には、名称変更したいテーブルのテーブルイメージを選択すると、パレットにそのテーブルのプロパティが表示されます。
2. "名前"エリアに新しい名前を入力します。
   新しい名前は即座に適用されます。

**注:** テーブルの命名規則についてはテーブルとフィールドの命名規則を参照してください。

## テーブルの削除

データベースからテーブルを削除できます。

この操作は4DのSQLエンジン、またはストラクチャーエディターから行えます。
4DのSQL文に関する詳細は、4D SQL Referenceマニュアル (MissingRef) を参照してください。

4Dでは2つの方法でテーブルを削除できます: 恒久的削除はテーブルおよびそのデータが実際にデータベースから取り除かれます。非恒久的削除ではテーブルが単にゴミ箱に置かれ (ゴミ箱ページ参照)、あとで元に戻すことができます。

### 恒久的削除

ストラクチャーエディターでテーブルを恒久的に削除するには:

1. 削除するテーブルを選択し、4Dの**編集**メニューから**クリア**を選択します。
   または
   テーブルのコンテキストメニューから**削除**を選択します。
   警告ダイアログが表示され、処理を続行するか確認されます:

**OK**をクリックすると、4Dは以下の処理を行います:

- テーブルは恒久的にストラクチャーから削除されます。またデータファイルから、そのテーブルのデータが恒久的に削除されます。
- テーブルのトリガーメソッドが削除されます。
- テーブルフォームはプロジェクトフォームとなり、エクスプローラーのゴミ箱に移動されます (ゴミ箱ページ参照)。

### 非恒久的削除

テーブルの非恒久的削除は、エクスプローラーのゴミ箱に、削除するテーブルを移動することで行われます。これはWindowsやMac OS実装されたファイル削除のメカニズムと同様のものです。

これを行うとテーブルは4Dのエディターに表示されず、その内容にアクセスすることもできなくなりますが、ゴミ箱を空にしない限り元に戻すことが可能で

す。

ストラクチャーエディターからテーブルをゴミ箱に移動するには:

1. テーブルのコンテキストメニューから**ゴミ箱に移動**コマンドを選択します。
   テーブルは即座にゴミ箱へと置かれます。
   または
   削除するテーブルを選択し、DeleteまたはBackspaceキーを押します。
   どちらの場合も、処理の続行を確認するダイアログが表示されます。
2. **OK**をクリックすると、テーブルはゴミ箱に移動されます。ゴミ箱を空にしない限りゴミ箱ページから元に戻すことができます。

#### 削除されたテーブルの番号

テーブルが削除されても、データベースの安定性を損ねることを避けるために、データベース中の他のテーブルの番号は変更されません。そのため例えば3つのテーブルがあるデータベースにおいて、それらのテーブル番号が2, 4 および 5であるといったケースがありえます。

恒久的に削除されたテーブルのテーブル番号は、新しくテーブルが作成されるときに再利用される点に留意してください。

## テーブルプロパティ

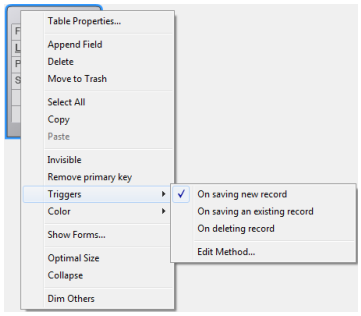
インスペクターパレットを使用してテーブルのプロパティを設定できます。また特定のプロパティについてはテーブルのコンテキストメニューからも設定できます。

### トリガー

**トリガー**はテーブルに関連する特定のイベントが発生した際に自動で実行されるメソッドです。以下のイベントがあります:

- On saving new record,
- On saving an existing record,
- On deleting a record.

トリガーイベントはインスペクターパレットまたはテーブルに割り当てられたコンテキストメニューを使用して設定できます:




有効にしたいトリガーにチェックします。

テーブルのトリガーには、ストラクチャーエディターのテーブルインスペクターパレットで**編集...**ボタンをクリックするか、テーブルのコンテキストメニューから**メソッド編集...**サブメニューを選択します。トリガーに関する情報は"4Dメソッドのタイプ"を参照してください。

**注:**

- エクスプローラーのメソッドページを使用してトリガーを表示および編集できます。
- ストラクチャーエディターの情報バーには有効にされているトリガーがテーブルごとに表示されます。

メソッドエディターに記述したトリガーは、選択したイベントが検出されたときにのみ実行されます。

### 属性

アピアランスや一般的な振る舞いを指定するために、テーブルごとに異なる属性を設定できます。

#### テーブルイメージのカラー

テーブルイメージごとにカラーを割り当てることができます。カラーを使用すると、大きなデータベーストラクチャーでグループを視認しやすくできます。例えば顧客や顧客にリレートするすべてのテーブルに同じカラーを設定し、在庫や在庫にリレートするすべてのテーブルに別のカラーを割り当てることができます。

**注:** 個々のフィールド (参照) やリレーション (定義参照) にカラーを割り当てることもできます。

テーブルにカラーを設定するには、テーブルを選択し、以下の方法でカラーを選択します:

- ストラクチャーエディターツールバーの**カラー**ボタン
- テーブルのコンテキストメニューの**カラー**コマンド
- インスペクターパレットの**カラー**オプション

**自動**オプションを使用すると、テーブルのデフォルトカラーが使用されます。

#### 非表示

このオプションを使用して、アプリケーションモードおよびプラグインからテーブルを見えなくできます。このオプションはユーティリティテーブルや、一時的に使用しないテーブルに対して使用できます。

テーブルを非表示にすることで、アプリケーションモードのすべてのエディター上やいくつかのダイアログボックスでそのテーブルとフィールドが非表示になり、ユーザーがテーブルに対して行うことのできる処理を制限できます。

アプリケーションモードの関連するエディターやダイアログボックスは以下の通りです:

- すべてのクエリーエディター (レコードの検索参照)
- 並び替えエディター
- ラベルウィザード
- クイックレポートエディター
- データの読み込みと書き出しダイアログボックス
- ダイアログボックス

これらのエディターそれぞれで、ユーザーはテーブルやそのフィールドを見たり選択したりすることはできません。例えばレポートやラベルに非表示テーブルのフィールドを含めることはできません。

**注:** これらのエディターを使用する際、ユーザーは設定をディスクファイルに保存することができます (例えば作成したクエリや並び替え設定)。その設定ファイルが使用される場合、設定が作成された後に非表示にされたテーブルやフィールドは、あとの操作でも使用することが可能です。さらに、ユーザーはダイアログボックスに非表示テーブルやフィールド名をタイプ入力することが可能です。

非表示テーブルとフィールドはストラクチャーエディターウィンドウ内でイタリックで表示されます。

#### レコードを完全に削除

このオプションを使用すると、DELETE SELECTIONコマンドを使用して選択したレコードを削除する際の動作を最適化できます。

4Dはレコードのセレクションを削除する際、レコードマーカーも削除します。レコードマーカーはそのレコードに関連する情報を含む、レコードに添付されたヘッダーです。マーカーとレコード両方を削除するのは、レコードだけを削除するよりも時間がかかります。特定のケースのためにレコードマーカーを自動では削除しないように設定できます。

このオプションを使用して希望する削除のタイプを設定できます。DELETE SELECTIONを使用して実行される大量のセレクションの削除を加速するために、**レコードを完全に削除**オプションの選択を解除できます。そうするとレコードマーカーは削除されなくなります。このオプションをプログラムで設定することはできません。

#### 複製を有効にする

このオプションがチェックされていると、4Dはテーブルのレコードを複製するために必要な情報を生成します (特にテーブルの主キーに基づきます)。レコードの複製機能は2つあるいはそれ以上の4Dデータベース間でのデータの同期を可能とし、セキュリティを高めます。

オプションを有効にしたら、特定の4DのSQLランゲージコマンドやHTTPプロトコルを使用して複製メカニズムを実装しなければなりません。このオプションと複製メカニズムに関する詳細はSQLを使用した複製を参照してください。HTTPを使用した複製に関する情報はURL 4DSYNC/を参照してください。

デフォルトで、このオプションはチェックされていません。このオプションを利用するには、複製するテーブルに対して主キーを設定しなければなりません。そうでなければこのオプションをチェックすることはできません。4Dはストラクチャーエディターで直接テーブルの主キーを設定できます。(後述)。

#### RESTサービスで公開

このオプションはそのテーブルが4Dデータベースへと送られるREST要求のコンテキストにおいて公開されるかどうかを管理します。デフォルトでは全てのテーブルがRESTに公開されています。
セキュリティ上の理由から、REST要求に対してデータベースの一部のテーブルのみを公開したい状況もあるでしょう。例えば、[Users]というテーブルを作成しその中にユーザー名とパスワードが保存されていたとしたら、そのテーブルは公開しない方が賢明でしょう。

テーブル(あるいはそれに含まれるフィールド全て)を公開したくない場合は、そのテーブルの**RESTサービスで公開**のオプションのチェックを外しましょう。

このオプションは "4D Mobile" 機能のコンテキストの中でも使用されます。詳細な情報に関しては、4D Mobile のマニュアルを参照して下さい。

#### ログファイルに含める

このオプションは、作成された全ての新しいテーブルと変換されたデータベースにおいては、デフォルトでチェックがされています。

テーブルのデータに対して行われたオペレーションを、全てデータベースのログファイル(作成時)に含めたい場合にはこのオプションにチェックを入れて下さい。このオプションは一般的に、ほとんどのテーブルにおいてチェックされるべきものです。しかしながら、最適化の観点から、例えばデータをインポートするために一時的に使用するテーブルなどではチェックを外すこともできます。

**注:** このオプションはテーブルがプライマリーキーを持っていない場合にはグレーになって選択することができません。

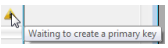
このオプションは、データベースがログファイルを使用する場合には、そのテーブルのデータがログに記録されなければならないという指示にすぎず、データベースレベルでのデータログ処理そのものを有効化するスイッチではないということに留意して下さい(詳細はログファイルの管理を参照して下さい)。

##### 警告メッセージ

必要要件を満たしていない場合は、**ログファイルに含める**のオプションの横に警告アイコンが表示されます。




このアイコンが表示されている限り、データログは有効化されていません。この警告にマウスオーバーすることで警告の原因を調べることができます:




以下のメッセージのいずれかが表示されます:

| メッセージ | 原因 | 修正箇所 |
|---|---|---|
| プライマリーキーが作成されるまで保留中です。 | テーブルにプライマリーキーがない場合はデータログを取ることはできません | SQL、またはテーブルのコンテキストメニューを使用してプライマリーキーを作成して下さい。 |
| *プライマリーキーの値が修正されるまで保留中です。* | プライマリーキーの値は検証され、異常が含まれています | フィールドのレコード内にて重複している値またはヌル値を全て除去して下さい。またはほかのプライマリーキーを使用して下さい |
| *「ログファイルに含める」が有効化されるまで保留中。* | データログを有効化するオプションがチェックされていません | データベース設定のバックアップ→設定ページ内の「ログファイルを使用」のオプションにチェックを入れて下さい |

### コメント

インスペクターパレットのコメントエリアにはテーブルに関する追加の情報を格納できます。このコメントはすべての開発者が利用できます。
フィールドおよびリレーションごとにもコメントエリアがあります。

### SQL

インスペクターパレットのSQLエリアにはスキーマメニューと情報エリアがあります。

- "スキーマ"メニューにはデータベースに定義されたすべてのSQLスキーマが一覧されます。このメニューを使用して、テーブルが属するスキーマを指定できます (このプロパティはSQLコマンドを使用して変更することもできます)。すべてのデータベースには最低1つの、DEFAULT_SCHEMAという名前のスキーマがあります。デフォルトですべてのテーブルはこのスキーマに属します。SQLスキーマに関する詳細は4D SQLリファレンスのスキーマを参照してください。
- 情報エリアは、テーブル名がSQLの命名規則に沿っているかどうかを表示します (例えば4Dと異なり、SQLではフィールド名にスペースを含めることはできません)。

### 主キーを設定、削除する

プライマリーキーは、テーブル内のレコードを個別に識別するために使用されるフィールドを指定します。4Dテーブルのレコードの複製(SQLを使用した複製を参照して下さい)とデータログ機能を使用する(ログファイルの管理を参照して下さい)ためにはプライマリーキーを設定しなければなりません。

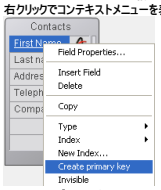
ストラクチャーエディターのコンテキストメニューを使用して、直接プライマリーキーを作成/編集/削除することができます。

**注:** プライマリーキーは、SQLランゲージにてPRIMARY KEY節のあとカラムのリストを記述することでも作成することができます(詳細な情報は *4D SQL Reference Guide* 内の主キーを参照して下さい)。

#### プライマリーキーの設定

ストラクチャーエディターからプライマリーキーを作成するためには、以下の手順に従って下さい:

1. テーブルのプライマリーキーを構成するフィールド(複数も可)を選択します。
2. 右クリックでコンテキストメニューを表示し、**主キーを作成**を選択します:




一つのテーブルにつき二つ以上のプライマリーキーを作成することはできません。プライマリーキーが既にテーブルにおいて指定されている場合、既存のプライマリーキーは無効化されるという旨の警告ダイアログボックスが表示されます。

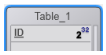
プライマリーキーは直ちに作成されます。プライマリーキーに含まれるフィールドはエディター上では下線が引かれ、インスペクターのSQLの詳細にはPRIMARY KEYのキーワードが入ります。

プライマリーキーに含まれるフィールドでは重複する値を含むことはできません。重複する値が既にテーブルのレコード内に存在していた場合、これのためにプライマリーキーを作成できないことを警告するダイアログボックスが表示されます。

**注**: プライマリーキーに所属するカラムは、ヌル値を受け付けません。

#### デフォルトのプライマリーキーフィールド

4D v14 以降、新しく作成されるテーブルには全てデフォルトでプライマリーキーが含まれています:




このフィールドは、デフォルトで名前が "ID" の倍長整数型のフィールドであり、以下の属性を持ちます:

- 重複不可
- ヌル値の入力を拒否
- 自動インクリメント
- 自動インデックス

**注:** 4Dアプリケーションの設定のストラクチャーページ にて、プライマリーキーのデフォルトの名前と型を変更することができます。

このフィールドはそのまま使用することもできますし、必要に応じて名前やプロパティを変更する事もできます ( 例えば UUID フィールドを使用する等 )。また、他のフィールドをプライマリーキーとして使用する場合はプライマリーキーを削除することはできます。

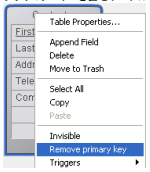
ただし、全ての 4D テーブルに最低一つずつプライマリーキーを用意しておくことが強く推奨されます。

**注:** デフォルトのプライマリーキーは、SQL CREATE TABLE コマンドを使用して作成されたテーブルやデータベースにインポートされたテーブルには追加されません。

#### プライマリーキーの削除

テーブルからプライマリーキーを削除するためには、以下の手順に従って下さい:

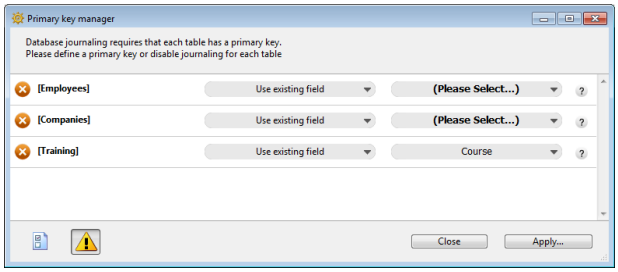
1. プライマリーキーを含むテーブルを右クリックしてコンテキストメニューを表示し、**主キーを削除**を選択します:




確認のダイアログボックスが表示されます。プライマリーキーを削除するためには**OK**をクリックします。

## プライマリーキー管理

「プライマリーキー管理」というアシスタントウィンドウは、特にデータログを使用しているデータベースにおいて、プライマリーキーがないテーブルに起因するエラーの解消を手助けするためのものです:




このアシスタントには以下の様な機能があります:

- データベースのそれぞれのテーブルと 4D のデータログのメカニズムとの競合性を検証します。
- データログのメカニズム合致しないテーブルに対してそれぞれに解決方法を提示します。具体的にはプライマリーキーの作成です。

**注:** プライマリーキー管理はv13でもコンポーネントとして提供されています。これを使用して v13 から v14 への変換の準備をすることができます。

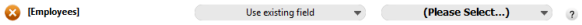
**アシスタントを表示**

プライマリーキー管理アシスタントを表示する方法は二通りあります:

- プライマリーキーエラーウィンドウから表示: **アシスタントを起動** ボタンをクリックしてアシスタントウィンドウを表示します。
  **注:** プライマリーキー管理はデザインモードで表示されます。データベースがアプリケーションモードで開かれた場合はアシスタントは直ちには表示されず、デザインモードへと切り替える必要があります。
- ストラクチャーエディターから: ウィンドウのツールバー内の新しいボタン をクリックするとアシスタントを表示できます。

**アシスタントの使用**

プライマリーキー管理ウィンドウではデータベース内の全てのテーブルに関しての情報が行となって表示されます:

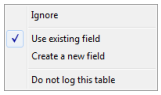



**注:** アシスタントはごみ箱に入っているテーブルは検知しません。

それぞれの行の先頭にあるアイコンはテーブルの状態をユーザーに知らせます:

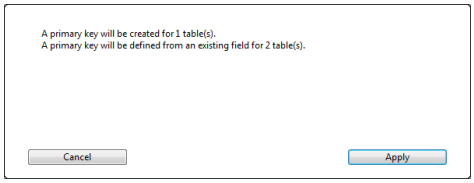
| | |
|---|---|
| OK | テーブルには有効なプライマリーキーがあります。 |
| プライマリーキーなし、使用可能なフィールド有り | テーブルにはプライマリーキーがないものの、プライマリーキーとして使用できるフィールドが少なくとも一つは含まれています。 |
| プライマリーキーも使用可能なフィールドもなし | テーブルにはプライマリーキーも、プライマリーキーとして使用できるフィールドもありません(プライマリーキーフィールドを作成する必要があります)。 |
| 警告 | テーブルにはプライマリーキーがありませんが、データのログを取りません(ログファイルに含めるのオプションのチェックが外されています )。この状態のテーブルは、警告アイコンをクリックすることによって非表示にできます。 |

データログを残すテーブルでプライマリーキーがないものに関して、アシスタントはそれぞれに実行すべきアクションを表示します。以下の選択肢から選ぶことができます:

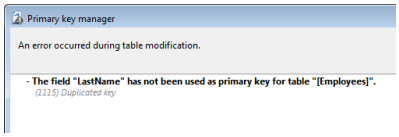



- **無視**: テーブルに何も変更を加えません。エラーは修正されず、テーブルの状態は変更されません。後でエラーを修正したい場合、またはアシスタントを使用せずにエラーを修正したい場合などにはこのオプションを選択して下さい。このオプションは特に、複数のテーブルをもとにプライマリーキーを作成したい場合に必要です。
- **既存のフィールドを使用** (プライマリーキーとして使用可能なフィールドが少なくとも一つはあるテーブルにおいてのみ表示):テーブルのフィールドのどれかをプライマリーキーとして指定します。このオプションを選択すると、アシスタントはデフォルトでプライマリーキーとして最適なフィールドを選択して提示します。
  他のフィールドを使用したい場合、またはアシスタントが適切なフィールドを識別するのに失敗した場合 ( この場合アシスタントは「選択可能なフィールド」を表示します ) などは、行の中の二つ目のメニューをクリックして選択可能なフィールドのリストを表示して下さい。
- **新規フィールドを作成**: テーブル内に新しいプライマリーキーを作成します。このフィールドは新規テーブルが作成されたときのものと同じ属性を持ちます(デフォルトのプライマリーキーフィールドを参照して下さい )。
  デフォルトでは、アシスタントは "ID" という名前の倍長整数型のフィールドが提示されます。このデフォルトのプライマリーキーの名前と型は環境設定の ストラクチャーページ から変更することができます。プライマリーキー管理の画面の中で をクリックすることにより、環境設定のページを開くことが出来ます。
  プライマリーキー管理ウィンドウ内で直接テーブルの名前/型を変更することもできます。
- **このテーブルはログに含めない**:
  このテーブルに関しては「ログファイルに含める」のチェックを外します。一時的なテーブルなどの場合にこのオプションを選択することができます (ログファイルに含めるを参照して下さい )。ダイアログボックスを確定させると、「警告」の状態がテーブルへと適用されできます。

設定が完了したら**適用 ...** をクリックして変更をデータベースに適用するか、データベースを変更せずに閉じる場合には**閉じる**をクリックして下さい。
**適用 ...** をクリックすると、実行されるオペレーションの一覧がダイアログボックスに表示され、適用かキャンセルがクリックできます。:




既存のフィールドをプライマリーキーとして使用した場合、4D はそれぞれのテーブルに関して、既存のデータがそのフィールドの型の固有性やヌル値の設定とあてはまるかどうかを検証します。もし、例えばフィールドが複製値やヌル値を含んでいた場合、エラーが生成されます:




データログを有効にするためにはこれらのエラーを探して削除する必要があります。

**注:** プライマリーキーに関するエラーの修正を容易にするために、プライマリーキーの値はMSCの アクティビティ解析ページ の中に表示されています。

## フィールドの作成と変更

テーブルごとに、格納および管理したいデータを保持するためのフィールドを作成する必要があります。

フィールドを作成する際、そこに格納される情報の種類を定義するフィールド型を割り当てます。4Dはフィールドの内容に対して異なる種類の処理を行うために、フィールド型を使用します。例えばフィールドが日付データを保持する場合、フィールドには日付型を割り当てます。そして4Dは期間の計算や日付の決定など、日付値の計算を行うことができます。さらに4Dは日付順でレコードを並び替えることができます。フィールド型については4Dのフィールドの型で説明しています。

フィールド型のほかに、テーブルのフィールドごとに、さまざまなプロパティーを設定できます。これらのプロパティはフィールドに対するデータの表示や更新、入力の条件を指定します。この点についてはフィールドプロパティを参照してください。

フィールド作成後、いつでもその型やプロパティを変更できます。またフィールドを削除することもできます。

### フィールドを作成する

1テーブルに対して32,767フィールドまで作成できます。標準の作成コマンド、あるいはコピー/ペーストを使用して新しいフィールドを作成できます。

フィールドを作成するには:

1. テーブル上で右クリックをし、コンテキストメニューから**フィールド追加**または**フィールド挿入**を選択します。
   または
   テーブルイメージの最後のフィールドの下の空の行の上をダブルクリックします。
   または
   テーブルを選択し、ストラクチャーエディターツールバーのオブジェクト追加メニューから、**フィールド**オプションを選択します:

   
2. (任意) 追加のフィールドを素早く作成するには、必要なだけ**改行**キーを押します。

フィールドがテーブルに追加され、4Dはそれを選択します。デフォルトでフィールドは*Field_X* (Xはフィールド番号に対応、ただしフィールドが削除されていた場合、この番号はテーブル中の実際のフィールド番号に対応しないかも知れません。後述の"削除されたフィールドの番号"参照) と名称がつけられます。

フィールド名やプロパティを設定/変更することができます (以下の段落参照)。

コピー/ペーストでフィールドを作成することができます:

1. 複製するフィールドを選択し、標準の"コピー"コマンド (4Dの**編集**メニュー、コンテキストメニュー、**Ctrl+C**/**Command+C**ショートカット)。
2. 新しいフィールドを作成するテーブルを選択し、**編集**メニューまたはコンテキストメニューから**ペースト**コマンドを選択するか、**Ctrl+V**/**Command+V**ショートカットを使用します。
   フィールドは同じ名称とプロパティで複製されます。対象のテーブルに同じ名称のフィールドが既に存在する場合、デーフォルトで"Copy_of_*FieldName*"という名称がつけられます。*FieldName*は元のフィールド名です。同じフィールドの複数のコピーを作成すると、番号が追加されます。

**注:** コピー/ペーストでフィールドが作成された場合、インデックスプロパティは保持されません。

### フィールドの名称変更

いつでもフィールド名を変更できます。4Dはそのフィールドが使用されているすべての場所 (フォーム、メソッド等) で名前を更新します。ただしフィールド名が更新されるときメソッドがエディタで開かれている場合、フィールド名の更新を有効にするにはいったんメソッドを閉じて開かなければなりません。

フィールド名はテーブル内でユニークでなければなりません。

ストラクチャーエディターやエクスプローラー、またはインスペクターパレットでフィールド名を更新できます。

フィールド名を直接更新するには (ストラクチャーエディターまたはエクスプローラーのテーブルページ):

1. フィールド名上でクリックを2回行います (既に目的のフィールドが選択されている場合、1回クリック)。
   または
   (ストラクチャーエディターのみ) 名称変更するフィールドを選択して、**Carriage return**を押します。
   編集モードに切り替わります。
2. 新しい名称を入力して、フィールドの外をクリックします。

ストラクチャーエディターのインスペクターパレットを使用してフィールドの名称を変更するには:

1. フィールド上をダブルクリックします.
   または
   フィールド上で右クリックをし、コンテキストメニューから**フィールドプロパティ...**を選択します。
   または
   エクスプローラーのテーブルページでフィールド名をダブルクリックします。
   4Dはインスペクターパレット上に選択したフィールドの名前とプロパティを表示します。パレットにはテーブル番号とフィールド番号も表示されます。インスペクターパレットが既に表示されていて、他のフィールドやオブジェクトが表示されている場合は、指定するフィールドを選択すると、パレット上にその情報が表示されます。
2. "名前"エリアに新しい名称を入力します。
   新しい名称はただちに適用されます。

フィールドの命名規則はテーブルとフィールドの命名規則を参照してください。

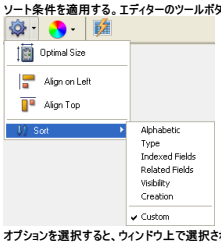
### フィールドのソート

ストラクチャーエディター中のテーブルに表示されるフィールドの表示順を変更することができます。テーブル中のフィールドの表示順を変更することで、、例えばストラクチャーの検証を容易にできます。

ストラクチャーエディター中でフィールドを並び替えても、アプリケーションの他のエディターには影響を与えません。実際4Dはフィールドを作成順 (**フィールド挿入**コマンドを使用した場合も含む) で表示します。ただしエクスプローラーなど一部のエディターでは文字の並び順で表示されます。

フィールドの表示順を変更するには2つの方法があります:

- フィールドをドラッグ＆ドロップで移動して、手作業で並び順を変更する。この場合、**Alt** (Windows) または**Option** (Mac OS) キーを押すとカーソルが手の形に変わり、フィールドを移動できるようになります:

  
  この方法で実行されたカスタムソートは記憶され、必要な時にソートメニューの**カスタム**コマンドを使用することで再現できます (後述)。
- ソート条件を適用する。エディターのツールボタンに割り当てられたソートサブメニュを使用して複数のソート条件を使用できます:
  
  
  オプションを選択すると、ウィンドウ上で選択されているテーブルすべてに適用されます。このメニューはテーブルが選択されていないと使用できません。
  現在適用されているソート条件には、メニューにチェックマークが表示されます。選択したテーブルに異なる条件が使用されている場合、複数のチェックマークが表示されます。
  選択できるソート条件は以下の通りです:
  - **文字コード順**: フィールドを文字順で表示します。
  - **タイプ**: フィールドを型毎に表示します (型ごとに文字順で表示されます)。
  - **インデックスフィールド**: インデックス付きフィールドを上に表示します。
  - **リレートしたフィールド**: 主キー、それから外部キーの順に表示します。
  - **表示設定**: 表示されるフィールドを上に表示します。
  - **作成**: 作成順にフィールドを表示します (デフォルト)。
  - **カスタム**: 特定のソートは適用されません。このオプションを選択すると、以前に手作業 (ドラッグ＆ドロップ) で設定された並び順が復元さ

れます。ドラッグ＆ドロップを行っていない場合、このオプションは効果がありません。
**注:** フィールドをテーブルに追加すると、カレントのソート条件に関わらず、それは常に既存のフィールドの後に追加されます。

### フィールドの削除

テーブルからフィールドを削除することができます。これは取り消すことができません。

この操作はSQLエンジンあるいはストラクチャーエディターを使用して4Dから行うことができます。

4DのSQL文に関する詳細は*4D SQL Reference*マニュアルを参照してください。

ストラクチャーエディターでデータベースから1つ以上のフィールドを削除するには:

1. 削除するフィールドを選択します。
   同時に異なるフィールドのフィールドを削除できます。
2. 4Dの**編集**メニューから**クリア**を選択します。
   または
   テーブルのコンテキストメニューから**削除**を選択します。
   または
   **Delete**または**Backspace**キーを押します。
   警告ダイアログボックスが表示され、処理を続行できます。
   **OK**をクリックすると、フィールドはストラクチャーから削除されます。そのフィールドのデータはデータファイルから削除されます。

#### 削除されたフィールドの番号

データベースの安定性を損ねることを避けるため、フィールドが削除されてもテーブル中の他のフィールド番号は変更されません。このため4つのフィールドがあるテーブルで、それぞれのフィールド番号が1, 4, 6そして8であるということがあり得ます。

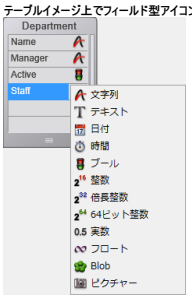
恒久的に削除されたフィールドの番号は、新しくフィールドが作成されるときに再利用される点に留意してください。

## 4Dのフィールドの型

フィールドごとにフィールド型を指定しなければなりません。フィールド型は、4Dがフィールドのデータをどのように処理・格納するか、またフォーム上でのデータの入力や表示に影響します。
デフォルトで新しいフィールドには文字型が割り当てられます。

フィールド型を変更するには:

1. テーブルイメージ上でフィールド型アイコンをクリックし、表示されるメニューから使用する型を選択します。




   または
   型を変更するフィールド上で右クリックし、コンテキストメニューのタイプサブメニューから新しい型を選択します。
   または
   型を変更するフィールドをインスペクターパレットに表示させ、タイプエリアのアイコンをクリックして型を変更します。

フィールド型は、フィールドにリレーションが設定されている、あるいは主キーに属する場合を除き、いつでも変更できます。これらのケースに該当する場合、フィールド型を選択するメニューは無効となります。フィールド型を変更するにはまずリレーションおよび主キーを取り除かなければなりません。
フィールドにデータを入力する前にフィールド型を変更した場合、4Dは単純にフィールド型を変更します。フィールドにデータを入力した後にフィールド型を変更した場合、4Dは可能であれば、データをロードする際にデータを新しい型に変換します。この変換がおこなわれても、そのレコードが更新されて保存されるまで、元のデータは保持されたままです。例えばテキストフィールドに"10以上"というデータが格納されていて、このフィールドの型を整数に変更すると、変更されたフィールドには"10"が表示されます。そのレコードを更新して保存をせずに、そのフィールドの型をテキストに戻すと、4Dは再び"10以上"を表示します。

ピクチャーフィールドを他の型に変換した場合、そのデータは表示されません。その逆も同様です。

デフォルトで4Dは以下のフィールド型をサポートします:

- 文字: 1から255文字長までの文字
- テキスト: 2 GBまでのテキスト
- 日付: 100年から32,767年までの日付
- 時間: 時:分:秒フォーマットの時間
- ブール: TRUEまたはFALSE日のみをとるフィールド
- 整数: -32,768から32,767の間の数値
- 倍長整数: +/-2,147,483,647の数値
- 整数64bit(1): +/- 2E64の8バイトの数値
- 実数: ±1.7E±308の間の浮動小数点数
- フロート(1): 浮動小数点数
- BLOB (Binary Large Object): 画像、他のアプリケーション、ドキュメントなどすべてのバイナリーオブジェクト
- ピクチャー: 4Dがサポートするネイティブフォーマットのピクチャー (4Dランゲージリファレンスのサポートされるネイティブフォーマット参照)

(1) これらの型は4DのSQLエンジンでのみ利用されます。これらのフィールドを4Dランゲージで使用すると、これらの値は内部的に実数に変換されます。

**注:** "文字列"という用語は文字あるいはテキストタイプを示し、"数値"は正数、倍長整数、整数64bit、実数またはフロート型を示します。

### 文字

文字フィールドには文字 (文字と数字)、句読点、そしてアスタリスク (*) やパーセント (%)、ハイフン (-) などの記号を格納できます。255文字長を超えない、テキストとして扱われるべき情報を格納する場合に文字フィールドを使用します。

文字フィールドには標準インデックスあるいはキーワードインデックスを割り当てることができます。インデックスについてはインデックスの作成と変更を参照してください。

文字は最も一般的なタイプです。典型的にこのフィールド型は名前、住所、電話番号、郵便番号などに使用します。データ入力時、文字フィールドはすべての文字、数字、句読点、記号を受け入れます。

郵便番号には2つの理由で文字フィールドを使用します: 数値フィールドは先頭の0を表示せず、また郵便番号にはハイフンが含まれます。数値フィールドタイプと文字フィールドタイプどちらを使用するかは、数値としての計算を行うか、または数値としての並び替えを行うかどうかで決定します。

文字フィールドに格納できる最大文字長として、1から255を設定できます。

2つ以上の文字フィールドを結合することができます。例えば人名の姓と名をラベルフォームに印刷したい場合などです。メソッドでは以下のように記述します:

```
FullName:=[Employees]FirstName+" "+[Employees]LastName
```

変数FullNameを表示したり印刷したりできます。あるいは情報の一部だけを別な変数にとりだすこともできます。

### テキスト

テキストフィールドはいくつかの点を除き、文字フィールドと同じです。

テキストフィールドには2GBまでの文字データを格納できます。一般的にテキストフィールドはコメントや説明など長い文章を保持するために使用します。

最適化のため、テキストフィールドの内容はレコードの外に保存することができます (フィールドプロパティの"レコードに格納"参照)。この場合フィールドに標準のインデックスを割り当てることはできません。文字フィールドと同様、テキストフィールドにはキーワードインデックスを割り当てることができます。インデックスについてはインデックスの作成と変更を参照してください。

詳細フォーム中で、テキストフィールドにスクロールバーを表示させることができます。印刷するレポートでは、すべての情報が印刷されるようテキストフィールドの大きさを (複数ページにわたって) 拡張できます。

データ入力中、テキストフィールドは基本的な編集機能 (スクロール、エリア中でのワードラップ、ダブルクリックでの文字の選択、矢印キーでの挿入ポイントの移動、そして標準のカット/コピー/ペースト処理) を提供します。複数行オプションが選択されていると、データ入力中テキストフィールドは改行を受け入れます (文字フィールドは受け入れません)。フォームオブジェクトプロパティでマルチスタイルオプションが設定されている場合、フィールドの内容を下線やボールド、フォントやサイズ、カラー指定などのスタイル付きテキストとして表示させることもできます。

テキストフィールドにワープロソフトなどからテキストをペーストできます。

**注:** テキストをレコードに格納する別の方法として4D Writeプラグインがあります。4D Writeでは、異なるフォント属性、段落揃えなど、標準のテキストフィールドでは利用できない他のワープロ機能を使用できます。他の4Dプラグインと同様、4D Writeエリアのデータはテキスト型ではなくBLOB型のフィールドに格納されます。4D Writeの利用に関する詳細は、4D Write Language Referenceマニュアルの4D Writeランゲージの紹介等を参照してください。

### 日付

開始日、購入日、誕生日などの日付データを格納する場合に日付フィールドを使用します。日付フィールドは100年から32,767年までの間の日付 (年/月/日) を格納することができます。

**注:** 言語設定が日本の場合、日付は年/月/日 (YYYY/MM/DD) のフォーマットで指定されます。他の言語設定では他のフォーマット (MM/DD/YYYYやDD/MM/YYYY) が使用されます。4DはコンピューターのOSに設定された言語のフォーマットに基づき、日付の入力を受け付け、表示します。

### 時間

現在時刻、開始時刻、継続時間など時間を管理する際に時間型を使用します。時間フィールドにはHH:MM:SSフォーマットで入力された時間を格納できます。

### ブール

ブールフィールド (論理フィールドと呼ばれることもあります) はTRUEまたはFALSE値を格納することができます。

ブールフィールドはチェックボックスあるいは2つペアのラジオボタンとしてフォーマットできます。チェックボックスではチェックされている状態がTRUE、されていなければFALSEです。ラジオボタンの場合、最初のラジオボタンが選択されていればTRUE、二番目が選択されていればFALSEです。

ブールフィールドには質問になるようなの名前を付けることができます。検索時には質問の答えがTRUEまたはFALSEでデータベース中のブールフィールドを検索できます。例えばブールフィールドに"Gender"ではなく"Male"と名前を付けます。この場合、"Gender is equal to TRUE"ではなく、"Male is equal to TRUE"という検索条件を書けます。

### 整数

整数フィールドには-32,768から32,767までの、小数点を含まない整数値を格納できます。

**倍長整数**

倍長整数フィールドは、整数型にはおさまらない整数値を格納するために使用します。小数を含まない ±2,147,483,647の値を格納できます。

**整数64bit**

この型のフィールドは8バイトを使用し、とても大きな整数 (+/- 2E64) を格納するために使用できます。

**注:** この型のフィールドは4DのSQLエンジンのみで使用されます。このフィールドが4Dランゲージで使用されると、内部的に実数に変換されます。

**実数**

実数フィールドには小数点を含む実数値を格納します。実数フィールドには小数を含む±1.7E±308の値を格納できます。
数値表示フォーマットは自動でシステムの地域パラメター設定が使用されます。4Dは千の桁区切りと小数点を、OSに設定された文字で置き換えます。

**フロート**

このフィールドは浮動総数点数を格納するために使用します。この型の数値は精度を失うことなく、実数値を格納することができます。

**注:** この型のフィールドは4DのSQLエンジンでのみ使用されます。このフィールドが4Dランゲージで使用されると、値は内部的に実数に変換されます。

**BLOB (Binary Large Object)**

Blob (Binary Large Object) フィールドはあらゆる種類のバイナリーデータを格納するために使用されます。例えば他のアプリケーションで作成されたドキュメント、スキャンしたピクチャー、あるいは他のアプリケーションを格納できます。BLOBには2GBまで格納できます。BLOBフィールドを含むレコードを処理する際、BLOB全体がメモリーにロードされます。

BLOBフィールドを使用してドキュメント管理システムを作成し、ユーザーのリクエストに応じてそれを配信するといったことができます。

4DのBLOBコマンドを使用してBLOBフィールドを管理できます。DOCUMENT TO BLOBとBLOB TO DOCUMENTコマンドを使用してBLOBフィールドにドキュメントを読み込んだり、ドキュメントを書き出したりできます。COMPRESS BLOB、EXPAND BLOB、およびBLOB PROPERTIESコマンドを使用して、BLOBを圧縮できます。BLOBの処理に関する詳細は、4D Language ReferenceマニュアルのDISABLE MENU ITEMを参照してください。

最適化のため、BLOBはレコードの外に保存されます。BLOBは必要な時にのみロードされます。例えば検索されたレコードを一件ロードする際などです。

BLOBはあらゆるタイプのデータを表現できるため、BLOBフィールドの内容をスクリーンに表示することはできません。

**ピクチャー**

ピクチャーフィールドにはグラフィックアプリケーションを使用して作成されたデジタル化された写真、図表、地図、イラストなどを格納できます。ピクチャーはネイティブフォーマットで格納されます。ピクチャーにはPostScript™や (4D v12以降は) メターデータなど出力デバイスに特別な指示を与えるための追加の情報が格納されていることがあります。この情報はピクチャーがピクチャーフィールドにコピーされる際に一緒に格納され、4Dが対応するデバイスでピクチャーを印刷する際、あるいはメタデータの場合はGET PICTURE METADATAやSET PICTURE METADATAで使用できます。

最適化のため、ピクチャーはレコードの外に保存されます。ピクチャーは必要な時にのみロードされます。例えば検索されたレコードを一件ロードする際などです。4D v13より、ピクチャーをデータファイルの外に保存できるようにもなりました (データをデータファイル外に保存参照)。

**ピクチャーファイルのデフォルト名**

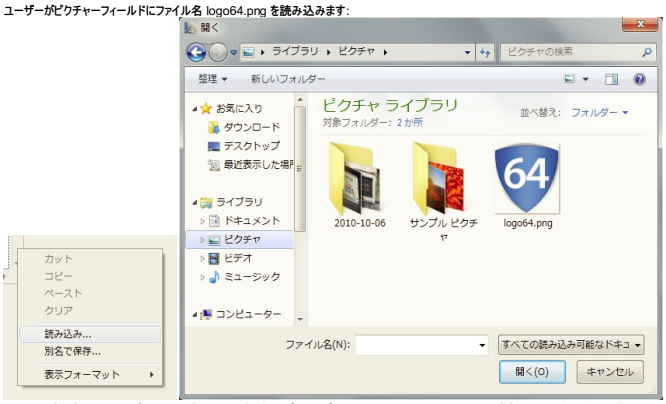
4D v13 ではフィールドに格納された各ピクチャーごとにデフォルト名を記録できます。つまりユーザーによる書き出しやWRITE PICTURE FILEコマンド等を (fileName 引数に空の文字列を渡して ) 使用してピクチャーフィールドの内容をディスクに保存する際に使用されるデフォルトのファイル名を設定できます。フィールドの内容を変数や他のフィールドにコピーした場合、デフォルト名もコピーされます。

ピクチャーフィールドに格納されるピクチャーのデフォルト名を設定する方法は2つあります:

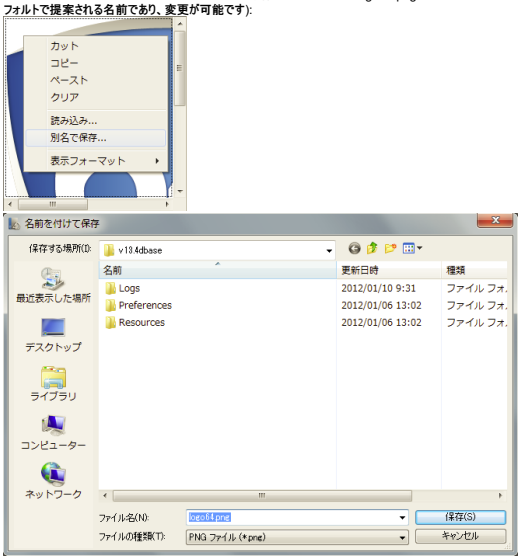
- 新しいコマンドSET PICTURE FILE NAMEを使用してプログラムで行う。このコマンドを使用してピクチャーのデフォルトファイル名を割り当てられます。また新しいコマンドGet picture file nameを使用してピクチャーのデフォルト名を取得することもできます。
- コンテキストメニューやREAD PICTURE FILEコマンドを使用してピクチャーフィールドにピクチャーファイルの内容を読み込む際は自動で設定されます。この場合 4D は元のピクチャーファイルの名前を記録します。

これは以下のように動作します:

1. ユーザーがピクチャーフィールドにファイル名 logo64.png を読み込みます:




2. 次にユーザーがレコードからピクチャーを書き出します。保存ダイアログには logo64.png というファイル名がデフォルトで表示されます (これはデフォルトで提案される名前であり、変更が可能です):



# フィールドプロパティ

名前と型に加え (4Dのフィールドの型参照)、フィールドプロパティもフィールドのアピアランス、データ入力、表示、編集やデータ格納条件などを規定します。テーブルの各フィールドのプロパティは個別に設定できます。

フィールドプロパティはインスペクターパレット、または特定のものについてはフィールド上を右クリックして表示されるコンテキストメニューで設定できます。

**注:** データが入力されているフィールドのプロパティの変更することができます。特定の変更は、既存のデータを考慮します。例えば、重複不可属性を有効にした際、すでに重複するデータが存在するならば、4Dは警告ダイアログを表示して、この属性を有効にするのを許可しません。また入力属性の変更によって、既存のデータが影響を受けることはありません。例えば、ヌル値の入力を拒否属性を有効にした場合には、その後に行われる入力のみチェックされます。すでに入力されている値に、空のフィールドを含むかもしれません。

## 定義

インスペクターパレットの"定義"エリアでは基本的なフィールドプロパティを設定します。いくつかのプロパティは特定の型のフィールドが選択されたときにのみ指定できます。

### カラー

フィールドごとにカラーを指定できます。カラーは役割や属性によってフィールドを見分けるために使用できます。例えば重複不可フィールドに特定のカラーを使用して、必須入力フィールドに別のカラーを指定するといったことができます。
テーブルごと (テーブルイメージのカラー参照) あるいはリレーションごと (リレーションプロパティの定義参照) 、個別にカラーを割り当てることもできます。

**注**: ストラクチャーエディターで設定されたカラーは、フォームに表示されるフィールドのカラーには影響ありません ( 背景および枠線の色参照)。

フィールドに対して設定されたカラーはフィールド名に適用されます。フィールドにカラーを設定するには、フィールドを選択し以下のいずれかを使用します:

- ストラクチャーエディターのツールバーの**カラー**ボタン
- フィールドのコンテキストメニューの**カラー**コマンド
- インスペクターパレットのカラーオプション

**自動**オプションを使用するとフィールドの標準オリジナルカラーが使用されます。

### 非表示

フィールドに対し**非表示**プロパティを設定すると、アプリケーションモードやプラグインに対し、フィールドを非表示にすることができます。**非表示属**性はユーザーからフィールドを隠します。この属性が設定されたフィールドはアプリケーションモードで表示される標準の4Dエディターやダイアログボックスに現れません。さらにプラグインから使用することもできません。アプリケーションモードの以下のエディターやダイアログボックスが関連します:

- すべてのクエリエディター(レコードの検索を参照して下さい)
- 並び替えエディター
- ラベル
- クイックレポートエディター
- データの読み込みと書き出しのダイアログボックス
- フォーミュラエディター

これらそれぞれの場所で、ユーザーはこのフィールドを見たり選択したりすることはできません。例えばクイックレポートエディターで、ユーザーは非表示フィールドを使用したレポートを作成することができません。

**注:** エディターを使用する際、ユーザーは設定をディスクファイルとして保存することができます (例えばクエリや並び替えの設定など)。この場合、設定が保存された後に非表示に設定されたフィールドは、この設定ファイルをロードした場合利用すること が可能です。さらにユーザーはフォーミュラエディターに非表示フィールドをタイプ入力することはできます。

ストラクチャーエディターウィンドウで、非表示フィールドはイタリックで表示されます。

### 重複不可

重複不可属性は、各々のレコードがそのフィールドに重複しない値を格納することを保証したい場合に使用します。重複不可属性は、テーブル中でレコードをユニークに識別するフィールドに使用します。 重複不可フィールドは例えば従業員番号や社会保障番号、注文番号などを検証する場合に便利です。

重複不可属性は空の値や入力された値などが重複して存在することを避けます。空のフィールドを他のレコードにコピーすることはできません。

ストラクチャーエディターの情報バーには、重複不可属性がフィールドに設定されているかどうかが表示されます (ストラクチャーエディター参照)。

### ヌル値の入力を拒否

**ヌル値の入力を拒否**プロパティはヌル値の格納を防ぐために使用されます。

この属性がフィールドに対して設定されると、そのフィールドにヌル値を格納することができなくなります。この属性が設定されたフィールドがヌル値を受け取ると、エラーが生成されます。

この低レベルなプロパティはSQLのNOT NULL属性に対応します。一般に4Dデータベースでヌル値を使用したい場合、もっぱら4DのSQLランゲージを使用することを推奨します。

**注:** 4Dにおいて、フィールドに"必須入力"属性を設定できます (後述)。これら2つのコンセプトは同様ですが、しかしスコープが異なります。"必須入力"属性はデータ入力制御であり、"ヌル値の入力を拒否"属性はデータベースエンジンレベルです。

### ヌル値を空値にマップ

このプロパティは、4Dランゲージでフィールドを扱う際に、空値とヌル値を等しく扱うために使用します。
互換性の理由で、4Dデータベーステーブルに格納されたヌル値は、4Dランゲージで処理される際に自動でデフォルト値に変換されます。例えば以下の文において:

```
myAlphavar:=[mytable]MyAlphafield
```

MyAlphafieldフィールドにヌル値が格納されていた場合、myAlphavarには空の文字列 "" が代入されます。
デフォルト値はデータ型により異なります:

- 文字およびテキストデータ型: ""
- 実数、整数、倍長整数データ型: 0
- 日付データ型: 00/00/00
- 時間データ型: 00:00:00
- ブールデータ型: False
- ピクチャーデータ型: 空のピクチャー
- BLOBデータ型: 空のBLOB

他方、この原則のメカニズムはクエリなど4Dデータベースエンジンレベルでの処理には適用されません。実際空値 (例えばmyvalue=0) を検索してもヌル値は見つからず、ヌル値を検索しても空値は見つかりません。両タイプの値 (デフォルト値とヌル値) が同じフィールドの異なるレコードに保存されている場合、追加のコーディングや特定の処理が必要となります。この不便さを解消するため、**ヌル値を空値にマップ**オプションを使用して4Dランゲージのすべての処理を標準化できます。このプロパティを使用すれば、デフォルト値を使用する原則がすべての処理に拡張されます。ヌル値を格納したフィールドは自動でデフォルト値を格納しているものとして扱われます。

このプロパティはデータベースエンジンの低レベルで適用されます。 特にIs field value Nullコマンドの動作に影響を与えます。

**注:** UUIDフォーマットの文字フィールドではUUIDが生成されていない場合でもNULLとは扱われません (UUIDフォーマット参照)。

### 自動インクリメント

"自動インクリメント"プロパティはレコードのユニークIDを生成するために使用されます。この属性は整数、倍長整数、整数64bit型 (4Dのフィールドの型参照) に使用できます。

このプロパティを持つフィールドは、テーブルにレコードが作成されるたびに、自動で値が増分されます。フィールドに代入される値は、4Dが管理するテーブルの"内部カウンター"に一致します。この内部カウンターには4DコマンドSequence numberとGet database parameter、および#Nマーカー (デフォルト値参照) を使用してアクセスすることもできます。

代入される数は永久的であり、レコードが削除されても再利用されることはありません。トランザクション中で生成された番号は、トランザクションがキャンセルされると失われます。
しかし、テーブルの内部カウンターはSET DATABASE PARAMETERコマンドを使用して再設定することができます。

"自動インクリメント"プロパティはSQL AUTO_INCREMENT属性に対応し、以下の例題のようにランゲージを使用して設定することもできます。この属性を持つフィールドのインスペクターにはSQL情報エリアにAUTO INCREMENTラベルが表示されます。

以下のコードは"animals"テーブルに整数型の"id"フィールドを作成します:

```
ARRAY TEXT($names;6)
$names{1}:="dog"
$names{2}:="cat"
$names{3}:="penguin"
$names{4}:="snake"
$names{5}:="butterfly"
$names{6}:="ostrich"
Begin SQL
   CREATE TABLE animals(
   id INT NOT NULL AUTO_INCREMENT ,
   name VARCHAR(30)NOT NULL,
   PRIMARY KEY(id));
```

```
    INSERT INTO animals(name)VALUES(:$names);
 End SQL
```

以下のコード:

```
SELECT * FROM animals;
```

は以下の値を返します:

| id | name |
|---|---|
| 1 | dog |
| 2 | cat |
| 3 | penguin |
| 4 | snake |
| 5 | butterfly |
| 6 | ostrich |

#### Stored in the record

このオプションメニューはテキスト、BLOB、そしてピクチャー型のフィールドで、フィールドデータを格納する場所を指定するために利用できます。以下のオプションから選択できます:

- **レコード**: データは各レコードとともに保存されます。大きなデータを扱う場合、通常このオプションを選択すべきではありません。ただしテキスト型の場合デフォルトでこのタイプが選択されています。これはBツリーインデックスを使用する場合にこのデータ格納タイプを選択していなければならないためです。テキストデータをレコード外に保存するよう設定すると、インデックス選択メニューが非表示になります。
- **データファイル**: データはデータファイル内に保存されますが、レコードブロックとは別なエリアに格納されます。BLOBとピクチャーフィールドではこのオプションがデフォルトで選択されています。テキスト、BLOB、そしてピクチャー型の大きなデータをレコードの外に格納することで、データベースの動作を最適化できます。
- **外部**: データは個別のファイルとして、.4DDファイル外に保存されます。このオプションについてはデータをデータファイル外に保存で説明しています。
  **互換性に関する注意:** このオプションは、これが選択された後新規に作成されたレコードで有効です。すでに存在するレコードについてはこのオプションは適用されず、この場合、そのテーブルは内部/外部にデータを保存する混合モードで動作することになります。既存のレコードにもこのオプションを適用したい場合、**レコードの強制更新**オプションを使用してデータを圧縮してください (圧縮ページ参照)。

#### 内部ストレージの最大サイズ

前節で説明したとおり、最適化の目的で、テキスト、BLOB、およびピクチャーフィールドのデータをレコードの外あるいはデータファイルの外に保存できます。

上記の設定を行った場合でも、パフォーマンスの理由から、適切なサイズに制限したうえで、データをレコードに保存するよう強制することができます。

この設定は**内部ストレージの最大サイズ (バイト)** オプションで行います。ここに入力されたサイズ (バイト単位) よりもサイズが小さいデータは格納場所の設定に関わらずレコード内に保存されます。例えばピクチャーフィールドでこの値を30,000バイトに設定した場合、20 KBのピクチャーはレコード内に保存され、40 KBのピクチャーは上記のオプションで設定した場所に保存されます。デフォルト値は0で、すべてのデータがレコードの外に保存されます。

#### UUIDフォーマット

このプロパティは文字型のフィールドで利用できます。このプロパティは、このフィールドがUUID識別子を格納するためのものであることを示します。格納されるデータはUUIDフォーマットを満たしていなければなりません (32の文字 (A-F, a-f) および数字 (0-9) の組み合わせ)。これを行うために自動UUIDプロパティ、Generate UUIDコマンド、あるいはカスタムアルゴリズムを使用できます。

このフィールドにUUIDフォーマットでない文字列を格納使用とすると、4Dはそれを自動で変換します。同じ処理は非文字フィールドからUUIDフィールドに変更された際にもその内容に対して適用されます。つまりレコードをロードする際、値は再フォーマットされ、表示されます。

**UUIDフォーマット**属性付きのフィールドをフォームに表示することができ、入力を行うことさえ可能です。文字は大文字で表示されます。小文字を表示したい場合、変数を使用する必要があります。

**注:**

- UUIDフォーマットのフィールドにキーワードインデックスや選択リストを割り当てることはできません。
- UUIDフォーマットの2つのフィールド間をリレーションで結ぶことができます。しかし標準の文字フィールドとUUIDフォーマットフィールド間にリレーションを作成することはできません。
- 一度初期化 (生成) されたUUIDフィールドがNULL値を持つ場合、それは空の文字列を返します。値が生成されていないUUIDフィールドはNULLではなく"000..." (0の数は文字数と同じ) を返します。プロパティは値が生成されていないUUIDフィールドには適用されません ("000..."が表示される)。

#### 自動UUID

このオプションは**UUIDフォーマット**プロパティを選択したときにのみ利用可能となります。

**自動UUID**プロパティを使用して、フィールドに自動でUUID番号を生成することができます。

この番号は以下のコンテキストで自動的に計算されます:

- レコードが作成されるとき。
- UUIDフィールドにNULL値が含まれるレコードがロードされるとき。このようなレコードは特に、**NULL値を空値にマップ**がチェックされていない状態で、UUIDフィールドがテーブルに追加される前に作成された場合に発生します。

どのような場合であれ、自動で生成されるUUIDをフィールドに保存するためには、レコードを保存しなければなりません。

**注:** データが読み込まれる場合は、このプロパティが選択されていても、4Dは新しい番号を生成せず、読み込んだ値を使用します (そしてフォーマットが有効でない場合は変換を行います)。しかし読み込んだ値が空の場合はUUIDが自動で生成されます。

#### QuickTime圧縮

**互換性に関する注意**: このオプションは以前のバージョンの4Dとの互換性の目的で保持されており、デフォルトでは何の効果もありません。これを使用可能にするためには、アプリケーション内でQuickTimeを有効化する必要があります(ピクチャを参照して下さい)。

#### テキストのクエリや並び替えでタグを使用しない

このプロパティはテキストおよび文字フィールドで利用できます。このオプションを選択すると、フィールドに格納されたデータに対するクエリーや並び替えはスタイルタグを考慮しなくなります。

このオプションはフォーム中のテキストエリアのデータに異なるスタイルを適用する4Dの能力に関連します。この機能に関する詳細はマルチスタイル (リッチテキストエリア)を参照してください。

スタイルの設定はテキストにHTMLタグを挿入することで行われます。これらのタグはテキストエリアが表示される際に解釈されます。

スタイルタグはデータとともに格納されます。例えばテキストフィールドに"week end"と入力すると、4Dは"week <SPAN STYLE = "color: #D81E05">end</SPAN>"を格納します。この処理はフォームレベルでユーザーに対し透過的です。しかしクエリーや並び替えにおいては、スタイルタグを無視するよう4Dに指示するために特別な設定が必要です。先の例で"week end"を検索するためには、巣と r 買う茶ーエディターでフィールドに対し**テキストのクエリや並び替えでタグを使用しない**オプションを選択する必要があります。

**注:** このオプションが設定されている場合、4Dで次のコードにより*thefield* から*thevalue*の値を検索するのと同じです。:

```
 QUERY BY FORMULA(OBJECT Get plain text(thefield)="thevalue")
```

### インデックス

**インデックス**プロパティはBLOBとピクチャーを除くすべてのフィールド型で利用できます。**キーワードインデックス**プロパティは文字、テキストおよびピクチャーフィールドで利用できます。

インデックスを使用すると、データ間の処理や検索の速度を向上させることができます。

インデックスの管理についてはインデックスの作成と変更で説明しています。

### データ入力制御

データ入力制御はフォームレベルでフィールドや入力可オブジェクトに対し設定できます。データ入力制御はフォーム上のフィールドや入力可オブジェクトに、ユーザーが入力できるものを制限します。

#### 必須入力

フィールドに**必須入力**属性が設定されていると、データ入力時にユーザーはそのフィールドに値を入力しなければなりません。4Dは必須入力フィールドが空である場合、そのレコードを受け入れません。必須入力はデータベース中で必須の情報を格納するフィールドに設定します。レコードをユニークに識別するためのフィールドは、必須入力を設定すべき良い例です。社会保障番号、請求番号、従業員番号などには必須入力属性を設定して、レコードの整合性を保護することができます。

この属性は特定のフォームのフィールドに対して設定することもできます。ストラクチャーエディターで**必須入力**属性を選択すると、フォームごとにそれを解除することはできません。他方、ストラクチャーエディターでこの属性が設定されていない場合に、フォーム上で**必須入力**属性を適用することは可能です。フォームでフィールドに必須入力属性を設定することに関しては、"入力可および必須入力属性を設定する"を参照してください。

**注:** 4Dにおいて、フィールドには"ヌル値の入力を拒否"プロパティを設定することができます (前述)。2つのコンセプトは同様ですが、スコープが異なります。"必須入力"属性はデータ入力制御であり、"ヌル値の入力を拒否"属性はデータベースエンジンレベルで動作します。

#### 修正不可

Iフィールドにこの属性が設定されていると、4Dはフィールドに最初に入力された値を受け入れますが、レコードが保存された後は、ユーザーがそれを変

更することを許可しません。ユーザーは最初にレコードが作成されるとき、レコードが保存される前までしか、入力を編集することはできません。レコードが保存されると、フィールドの値は編集できなくなります。その後、値はメソッドあるいはデザインモードでこのオプションを取り除いた後でのみ、編集可能です。

修正不可は会計監査の支払日等を格納するフィールドに使用されます。修正不可はしばしばテーブル中のレコードをユニークに識別するためのフィールドに使用されます。

**注:** この属性はページモードで詳細フォームに表示されるフィールドに対して動作します。他のケース (リスト入力、リストあるいはページモードのサブフォーム中への入力) では、フィールドの値は編集可能です。

#### 表示のみ

**表示のみ**が設定されたフィールドには、ユーザーはキーボードから値を入力することはできません。そのようなフィールドにはデフォルト値を設定するか、フィールドに値を挿入するコードを書かなくてはなりません。表示のみ属性を持つフィールドは、計算された総計やメソッドで割り当てられて一連番号など、ユーザーに編集させたくない値を表示する際に便利です。

特定のフォームに対して、フィールドを入力不可に設定することもできます。フォーム上のフィールドを入力不可に設定する方法は、"入力可と必須入力属性を設定する"を参照してください。

#### 複数行

このオプションはテキストフィールドで利用できます。このオプションがチェックされていると、そのあとに作成されたフォームで、テキストフィールドは自動で複数行のテキストを受け入れるよう設定されます。この場合のデフォルトの特徴は以下の通りです:

- 複数行に対応した高さ
- 水平スクロールバー
- 実行時、改行キーにより改行が挿入される

このオプションがチェックされていないと、フォーム上のテキストフィールドのデフォルトアピアランスは文字フィールドと同じです。一行分の高さでスクロールバーがなく、改行キーによりフォーム中の次のオブジェクトに移動します。

フォームエディターでこれらのアピアランスについてはいつでも変更できます。

#### 項目選択を使用する

**項目選択を使用する**属性を使用すると、フィールドに情報を入力する際、選択リストが表示されます。この属性を使用するためには、まずリストエディターを使用して選択リストを作成する必要があります (リスト参照)。

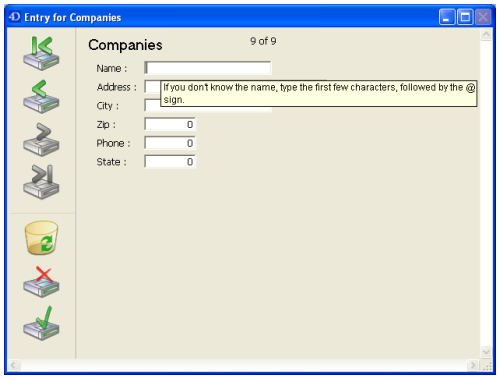
特定のフォーム上でフィールドに対し選択リストを割り当てることができます。しかしフォーム上だけで選択リストを割り当てると、そのリストはクエリーエディターなど他のエディターやダイアログボックス上では表示されません。フォームで選択リストを使用する方法についてはデータ入力の制御と補助を参照してください。

**注:** フォームレベルで指定リストや除外リストを設定することもできます。項目選択を使用するオプションをチェックしたら、メニューから作成済みの選択リストを選択します。[...]ボタンをクリックすると、直接リストエディターにアクセスできます (リスト参照)。

### ヘルプTip

フィールドにヘルプTipを追加して、ユーザーにフィールドに関する追加の情報を提供することができます。TipがヘルプTipエリアに入力されると、ユーザーがフォーム上でそのフィールド上にカーソルを合わせるとTipがフィールドの隣に表示されます。ヘルプTipはすべてのプラットフォーム上で表示されます。

ユーザーがマウスポインターをフィールド上に重ねると、tipは以下のように表示されます:




特定のフォーム上でTipを使用することもできます。特定のフォームのみにTipを割り当てると、それは他のフォーム上では表示されません。フォームにTipを割り当てる方法についてはヘルプメッセージを参照してください。

### コメント

インスペクターパレットのコメント欄はフィールドに関する追加情報を格納するために使用できます。このコメントはすべてのデベロッパーに対して表示されます。
テーブルやリレーションも各々コメントエリアを持っています。

### SQL

インスペクターパレットのSQLエリアには、SQLランゲージからそのフィールドを参照する際の有用な情報が表示されます。

このエリアには特に、指定された名前がSQLの命名規則に沿っているかどうかが表示されます (例えば4Dと異なりSQLは名前にスペースが含まれることを許可しません)。

フィールドでは、SQL情報エリアにはSQL属性も表示されます (型やプロパティ)。

## データをデータファイル外に保存

4D v13 では各 BLOB、ピクチャー、テキストフィールドごとにデータの保存先を指定できます。既存の保存オプション ( レコード内、またはレコード外 ) に加え、データファイル外への保存を選択できます。この場合データはディスク上の外部ファイルとして保存されます。外部ファイルはサードパーティアプリケーションで変更することができますが、この場合データ整合性は開発者が管理しなければなりません。

この動作はユーザーに対し透過的です。データの場所に関わらず、データへのアクセスは同じです。

ファイルの外に保存できることで、アプリケーションの動作を最適化することができます。例えば大量のデータを外部に保存できればその分データファイルのサイズを減らすことができます。またデータベースが開かれていないときでも OS からそれらのファイルにアクセスすることができます。

### 自動モード/カスタムモード

データファイル外部への保存は 2 つのモードで実行できます :

- **自動モード** : このモードでは、4D は特定の構造のフォルダーを自動で作成・管理し、すべてのデータをそこに格納します。この場合データは透過的に管理され、フィールドデータがデータファイル内部に格納された
場合と同様に動作します。
この自動モードを有効にするには以下のいずれかの方法を取ります :
  - ストラクチャーエディターで外部への格納を設定する (設定はデータベースストラクチャーに保存されます)。このオプションについてはStored in the recordで説明しています。
  - SET EXTERNAL DATA PATHコマンドのpath引数に空の文字列を渡す (設定は 4D を終了するまで有効)。
- **カスタムモード**: このモードでは各フィールドおよび各レコードごと、外部ファイルの保存場所を自由に選択できます。この場合 4D はフィールドとデータの間のリレーションのみを管理し、データベースメカニズムの一部は動作しなくなります。
カスタムモードはSET EXTERNAL DATA PATHコマンドを使用して、path引数にデフォルト以外のフォルダーを指定することで有効にされます。

以下の表では 自動およびカスタムモードの機能を比較します:

| | 自動モード | カスタムモード |
|---|---|---|
| 格納先フォルダーの設定 | 名前と場所は 4D が設定します。データベース全体で 1 つのデフォルトフォルダーを使用します。 | 名前と場所を自由に設定可能。各フィールドごとに変えることもできます。 |
| 外部ファイルの作成、ロード、保存 | 自動 | 自動 |
| レコードが削除された際の外部ファイルの削除 | はい | いいえ |
| フィールドに NULL 値が割り当てられたときの外部ファイルの削除 | Yes | いいえ |
| データベースが保存された際の自動統合 | はい | いいえ |
| ログファイル統合の際の自動サポート | はい | いいえ |
| 標準インデックスのサポート (テキストフィールド) | いいえ | いいえ |
| キーワードインデックスのサポート (テキストとピクチャー) | はい | いいえ |
| トランザクションのサポート | はい | いいえ |

### 外部データの場所

データファイルの外に保存されたデータは以下の原則に基づき整理されます:

- 各レコードごと、データは xxx.txt ( テキスト )、xxx.blob (BLOB)、xxx.jpgや xxx.tiff ( ピクチャー、拡張子はピクチャータイプに基づく ) という名前で保存されます。xxx は 4D が管理する UUID です。
- 自動モードでは、すべての外部データが <DatabaseName>.ExternalDataという名前のフォルダーに配置されます (DatabaseName はデータベーストラクチャーの名前です )。このフォルダーはデータファイル (.4DD) と同階層に作成されます。
4D はこのフォルダー内に、外部にデータを格納するフィールドを持つテーブルごとにフォルダーを作成し ( 名前は "Table"+ テーブル番号 )、さらに各外部フィールドごとにサブフォルダーを作成します (名前は"Field"+ フィールド番号)。フィールド中の最初の 100 項目はフォルダーの第一レベルに格納されます。それ以降は "2" から始まるサブフォルダーが順次作成され、その中に 100 要素までが格納されます。例えば"Accounting" データベースのテーブル番号 3 のフィールド番号 5 のデータがデータファイルの外に格納される場合、以下の構成でフォルダーやファイルが 4D により作成されます:
**Accounting.4DD**
**[Accounting.ExternalData]**
  **Table3**
    **Field 5**
      **Data_1B7F3A 56F6544B45951EFA60426D5ABC.txt**
      **Data_1B7F3A 56F6544B45951EFA60426D5CCC.txt**
      **...**
      **2**
        **Data_2ADBFBA478AAE4409DA9C2D13C90A53B.txt**
        **Data_32F8A30B87EE7E4BBC802468D553DC43.txt**
        **...**
- カスタムモードではSET EXTERNAL DATA PATHコマンドを使用して、各フィールドごと個別にフォルダーの名前と場所を自由に設定できます。この設定はデータベーストラクチャーには保存されません。

### ファイルの作成と更新

フィールドの外部ファイルへの記録は、レコードがディスクに保存される際に行われます (必要であればトランザクションが有効にされた後):

- 外部ファイルが存在しなければ作成されます。
- 外部ファイルが既に存在する場合、4D はそれを新しいファイルで置き換えます。
既存のファイルを保持したい場合、(SET EXTERNAL DATA PATHコマンドを使用して ) 異なるパスを指定するか、レコード保存前に既存の外部ファイルからその内容をメモリにロードするためにRELOAD EXTERNAL DATAコマンドを使用します。この方法はレコードがロードされた後、外部ファイルが他のアプリケーションによって変更された場合にも有効です。

#### 同期と複製

データの格納場所は各データベースごとのローカルパラメーターです。データの同期や複製を行う場合、これらのパラメーターはローカルデータベースとリモートのそれとでは異なるかもしれません。この場合データの格納は各データベースのパラメーターに従います。同期や複製は設定を変更しません。

例えばリモートデータベースにおいてピクチャーフィールドが外部ファイルとして保存され、同じフィールドがローカルデータベースではデータファイル内に保存されている場合、複製が行われると、ローカルデータベースのこのフィールドに追加されたすべての (データファイル内の) データは、リモートデータベースのデータファイル外に保存されます。

### ファイルへの外部からのアクセス

外部に格納されたファイルは 4D 以外のアプリケーション (OS、テキストやグラフィックエディター等 ) から読み書きモードでアクセスできます。しかしこれはアプリケーションの動作を変更してしまうかもしれないので、注意して行わなければなりません:

- 外部ファイルが OS やその他のアプリケーションによって削除、名称変更、あるいは移動されると、4D は対応するフィールドが NULL 値を持つものとして扱い、レコードが保存されるときに ( 値が NULL でなければ ) 新しいファイルが作成されます。
- インデックスを使用している場合、サードパーティアプリケーションが外部ファイルを変更しても、親レコードがディスクに保存されるまでは、インデックスが更新されません。

**注:** 外部テキストファイルは BOM なしの UTF-8 で保存されます。サードパーティアプリケーションでそのファイルを開き、BOM 付きで保存された場合、そのファイルを 4D で開くことはできますが、保存の際に BOMは失われます。

"読み込みのみ" モードでレコードをロードしても、このレコードの対象フィールドの外部ファイルはロックされません。ディスクファイルはその内容が 4D のメモリに読み込まれていますが、4D やその他のアプリケーションを使用して編集が可能です。

## テーブルとフィールドの命名規則

テーブルおよびフィールド名は以下の規則に沿っていなければなりません:

- 名前は31文字まで含めることができます。
- アルファベット文字で始まらなければなりません。
- 一文字目以外は文字、数字、スペース、アンダースコアを組み合わせることができます。
- 以下の文字をテーブル名として使用することはできません (ストラクチャーエディター上でエラーが生成されます): ( ) + - / * " ; = & | # > < ^ ' { } % DIAMOND (0x00D7), CUBE (0x00B3), SQUARE (0x00B2), PLUS-MINUS (0x00B1)
- 一般規則として、4Dや外部の言語が解釈を誤る原因となるような文字 (例えばコンマやコロンなどの句読点) は避けなければなりません。
- 4Dは31文字を超えた分に関しては切り取りを行います。そして名前の先頭と終わりのスペースを取り除きます。
- 2つのオブジェクトに同じ名前を使用してはいけません。同じ名前を2つ使用すると、すでにその名称が使用されていて入力が受け入れられない旨を警告するダイアログボックスが表示されます。
- テーブルやフィールド名に予約語を使用しないでください。予約語にはコマンド名 (Date, Time等)、キーワード (If, For等) そして定数が含まれます。

**注:** オブジェクトをSQLで処理する場合には追加のルールが適用されます。文字_0123456789abcdefghijklmnopqrstuvwxyzのみが受け入れられます。そして名前にはSQLキーワード (コマンド、属性等) が含まれていてはなりません。インスペクターパレットの"SQL"エリアには、名前がSQLの規則に沿っていない場合警告が表示されます (UNREGISTER CLIENT参照)。

**Tip:** オブジェクト名に空白を含めることが可能だとしても、そのようにすることはお勧めしません。空白なしで名前を付ければ、メソッドエディター中でダブルクリックでそのオブジェクトを選択できます。空白の代わりにアンダースコアをお勧めします。

# インデックスの作成と変更

頻繁に検索や並び替えで使用されるフィールドに、インデックスを設定できます。例えば姓、会社名、製品名などフィールドを使用してレコードの検索や並び替えを行う場合、これらのフィールドにインデックスを設定することができます。またこのプロパティはテーブル間でのリレーションを確立するためにも使用されます。この点に関する詳細はリレーションの作成と変更を参照してください。

フィールドにインデックスが設定されると、4Dはそのフィールド用に内部的なインデックステーブルを作成します。このテーブルを使用して、4Dはそのフィールドに対する素早い検索や並び替えを実行します。インデックスが設定されていないフィールドを使用して検索や並び替えを行う場合、4Dはシーケンシャルにデータをロードし、レコード毎順番に評価します。インデックスを使用すれば、4Dはすべてのレコードを読み込むことなく検索や並び替えを行えます。

インデックスは文字、テキスト、日付、時間、整数、倍長整数、整数64bit、実数、フロート、そしてピクチャー型に設定できます。レコードの作成、変更、削除を行うと、4Dは自動でそのインデックステーブルを更新します。既に存在するフィールドにインデックスを作成すると、4Dは自動で既存のデータをインデックス化します。必要なだけフィールドにインデックスを設定できます。

すべてのフィールドにインデックスを設定するようなことはしないでください。インデックスはデータベースのサイズを増大させ、ディスク上のスペースを消費します。またインデックスを設定すればするほど、レコードの保存に時間を要するようになる点に留意してください。4Dはレコードを受け入れるたびにインデックステーブルを更新しなければならないためです。

データベースを上位バージョンに変換する際やデータ圧縮時など、特定の処理を行った際にもインデックスが再構築されます。

ストラクチャーエディター中で、インデックスが設定されたフィールドは**太字**で表示されます。

### インデックスのタイプ

4Dでは3種類のインデックスタイプを提供しています。そのいずれを選択するかは一般的に期待する結果と、フィールドに現れるデータに基づきます。各インデックスタイプの説明は以下の通りです:

- **標準インデックス**: これは1つのフィールドで構成されるインデックスで、標準のデータベース処理 (検索や並び替え) の速度を向上するために使用します。4Dではさらにこのタイプのインデックスに対し、内部的な構造が異なるBツリーまたはクラスターBツリーを選択できます。
- **復号インデックス**: このインデックスは、しばしばともに検索条件に使用される (例えばLastName+FirstName) 2つ以上のフィールドの合成値を格納します。
- **キーワードインデックス**: このインデックスは文字、テキストおよびピクチャーフィールドで使用できます。テキスト内部の検索を容易にする目的で使用されます。またピクチャーの場合は、ピクチャーに割り当てたキーワードで検索を行うために使用できます。

#### 標準インデックス

標準インデックスはデータベース処理の速度を向上する目的で使用されます。4Dは2つのタイプの標準インデックス、BツリーとクラスターBツリーを提供します。

- **Bツリー**: 標準のBツリータイプインデックス。この汎用目的のインデックスはほとんどのインデックスの要件を満たします。
- **クラスターBツリー**: クラスターを使用したBツリータイプのインデックス。このアーキテクチャーはインデックスに含まれるキーの量が少ない場合 (例えば同じ値がデータ中に繰り返し現れる場合) に有効です。

インデックスアーキテクチャーの選択を容易にするために、**自動**オプションを選択できます。この場合4Dは自動でデータのタイプに応じインデックスのタイプを選択します。

**注:** テキストタイプのフィールドに割り当てたBツリーインデックスは、フィールドの最初の1024文字 (最大) を格納します。そのため、1024文字を超える検索を、インデックスを使用して行うと失敗します。

#### 復号インデックス

復号インデックスでは、レコード毎に2つ以上のフィールドの値の合成値を格納します。よく見られる例はFirstName+LastNameフィールドの復号インデックスです。この場合、"Peter Smith"の検索は (まず"Smith"を検索し次に"Peter"を検索する) 標準の検索よりも速くなります。

4Dは検索や並び替えの際に自動で復号インデックスを利用します。例えば復号インデックス"City+ZipCode"が存在するとき、"lastname=carter & city=new york & zipcode =102@"のような検索を行うと、自動でその復号インデックスが利用されます。
復号インデックスはインデックス作成ダイアログで作成します。このダイアログボックスについては後述の"インデックスの作成"を参照してください。

#### キーワードインデックス

キーワードインデックスは文字、テキスト、およびピクチャーフィールドに使用することができます。

- 文字およびテキストフィールドでは特別なインデックス、キーワードインデックスを使用すると、フィールドに格納されたテキストは単語ごとにインデックス化されます。1文字や2文字の単語も含め、すべての単語がインデックス化されます。このタイプのインデックスはテキストフィールドのキーワードによる検索速度を劇的に向上させます。
  文字やテキストフィールドには標準のインデックスとキーワードインデックス両方を割り当てることができます (データがレコード内に保存されている場合)。4Dはコンテキストに応じて適切なインデックスを使用します。

- ピクチャーフィールドにキーワードインデックスを使用すると、ピクチャーに割り当てられたキーワード (メタデータ) に対する検索が劇的に速くなります。
  警告: ピクチャーキーワードインデックスは**IPTC/Keywords**タイプのメタデータを使用します。これらのタイプのメタデータは特にTIFFとJPEGでサポートされています (BMP、PNG、GIFではサポートされていません)。他のタイプのメタデータはインデックスでサポートされません。
  ピクチャーのキーワードインデックスは、ピクチャーフィールドが保存 (レコードが作成・更新される、ピクチャーが読み込まれるなど) されるたびにIPTC/Keywordsタイプのキーワードがあれば、4Dが自動で更新します。SET PICTURE METADATAを明示的に呼び出す必要はありません。

キーワードインデックスに含まれるキーワードのリストを取得するためにDISTINCT VALUESコマンドを使用できます。

テキストやピクチャーフィールドのキーワードインデックスは % 演算子で使用します。インデックスの値を明示的に使用するには、この演算子をクエリまたは並び替えフォーミュラで使用しなければなりません。例えば:

```
QUERY([PICTURES];[PICTURES]Photos %"cats")
 // cats キーワードを持つ写真を探す
```

この書式はすべてのクエリおよび並び替えコマンドで同様に動作します : QUERY BY FORMULA, QUERY SELECTION, ORDER BY等。
% 演算子の動作については 4D ランゲージリファレンスマニュアル 比較演算子 を参照してください。

### インデックスリスト

ストラクチャーエディターのツールバーにある をクリックすると、インデックスリストウィンドウが表示されます。このウィンドウはタイプに関わらず、ストラクチャのすべてのインデックスを表示します:

ボタンはインデックスプロパティダイアログボックスを表示します。
ボタンは選択したインデックスを削除します(確認ダイアログが表示されます)。このボタンは特に複合インデックスを削除するために使用されます。
ツールボタンに割り当てられた2つのメニューを使用できます (インデックスが選択されている場合にのみ使用できます):

- **編集**: インデックスプロパティダイアログボックスに選択したインデックスのプロパティを表示します (次段落参照)。このダイアログはリスト中のインデックスをダブルクリックすることでも表示できます (名前エリアを除く)。
- **再構築**: 選択したインデックスを再構築するために使用されます。このコマンドを選択すると、確認ダイアログボックスが表示されます。

### インデックスの作成

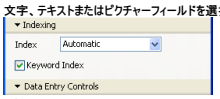
インデックスの作成方法はタイプにより異なります。さらにインデックスは直接あるいはダイアログを使用して作成できます。

**標準のインデックス**を直接作成する:

1. フィールドを選択し、インスペクターパレットのインデックスメニューから選択します。
   または
   フィールドを右クリックし、コンテキストメニューの**インデックス>**サブメニューから選択します:
   - Bツリー: 標準のBツリータイプのインデックスを作成します。
   - クラスターBツリー: クラスターを使用するBツリータイプのインデックスを作成します。
   - 自動: 関連するデータに基づき、4Dがインデックスのアーキテクチャーを決定します。
   - なし: インデックスを設定しないか、既存のインデックスを取り除きます。

**キーワードインデックス**を直接作成する:

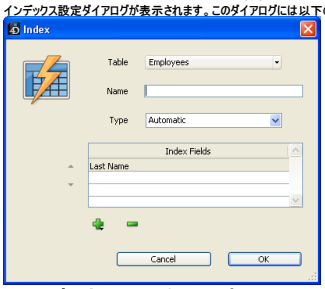
1. 文字、テキストまたはピクチャーフィールドを選択し、インスペクターパレットの"キーワードインデックス"オプションにチェックを入れる。




   または
   フィールドを右クリックし、コンテキストメニューから**インデックス>キーワード**を選択します。

**複合インデックス**を作成、あるいは他のタイプのインデックスをインデックス作成ダイアログで作成するには:

1. フィールドを選択し、ストラクチャーエディターツールバーのオブジェクト追加メニューを使用して、**インデックス**オプションを選択します。
   または
   フィールド上で右クリックし、コンテキストメニューから**新規インデックス...**を選択します。
   インデックス設定ダイアログが表示されます。このダイアログには以下の要素が含まれます:




   - **テーブル**: データベース中のすべてのテーブルのリスト。このメニューから、インデックスが属するテーブルを選択します。
   - **名前**: インデックス名入力エリア。この名前は4Dランゲージコマンドで使用されます。
   - **タイプ**: 作成するインデックスのタイプを選択するメニュー。"自動"を選択すると、フィールドの内容に基づき、4Dが自動でインデックスタイプを決定します。
   - **フィールドリスト**: このエリアで、インデックスに割り当てるフィールドを指定します。ストラクチャーエディターでフィールドを選択していると、あらかじめそのフィールドがこのリストに設定されます。

インデックスにフィールドを追加するには ボタンをクリックします。選択したテーブルのフィールドリストが表示され、インデックスに追加するフィールドを指定できます。

- 複合インデックスを作成するには、インデックスに含めるフィールドを複数選択します。フィールドリストが完成したら、矢印ボタンやドラッグ&ドロップを使用してフィールドの並びを変更できます。
- "キーワードインデックス"タイプは文字やテキストフィールドタイプでのみ使用できます。またこの場合、1つのフィールドのみをインデックスに含めることができます。

インデックスからフィールドを削除するには、リスト中でそれを選択し、ボタンをクリックします。
インデックスの設定ができたら、**OK**ボタンをクリックするとインデックスが生成されます。

**ストラクチャーエディターでインデックスを削除する**

利用しなくなったインデックスはいつでも削除できます。削除はストラクチャーエディター上で直接、またはインデックスリストウィンドウ上で行うことができます。インデックスリストウィンドウに関する詳細は、前述の"インデックスリスト"を参照してください。

標準インデックスを削除するには:

1. 削除するインデックスが割り当てられているフィールドを選択し、インスペクターパレットのインデックスメニューから**なし**を選択します。
   または
2. 削除するインデックスが割り当てられているフィールド上で右クリックし、コンテキストメニューから**インデックス>なし**を選択します。

キーワードインデックスを削除するには:

1. 削除するインデックスが割り当てられているフィールドを選択し、インスペクターパレットの"キーワードインデックス"オプションのチェックを外します。
   または
2. 削除するインデックスが割り当てられているフィールド上で右クリックし、コンテキストメニュー中の**インデックス>キーワード**を選択してチェック状態を解除します。

複合インデックスの削除はインデックスリストウィンドウのボタンを使用してのみ削除できます。

**インデックスの再構築**

フィールドのインデックスをいつでも再構築できます。これは言い換えると、そのインデックスに対応するインデックステーブルを、現在のデータを使用して再構築することを意味します。これはアプリケーションメンテナンスの際に利用できます。

インデックスの再構築はインデックスリストウィンドウの**再構築**コマンドを使用して行います。

データ言語の変更 (テキスト比較参照) や、データ圧縮 (GET STYLE SHEET INFO参照) などのメンテナンス処理においても、インデックスの再構築が行われることに留意してください。

## リレーションの作成と変更

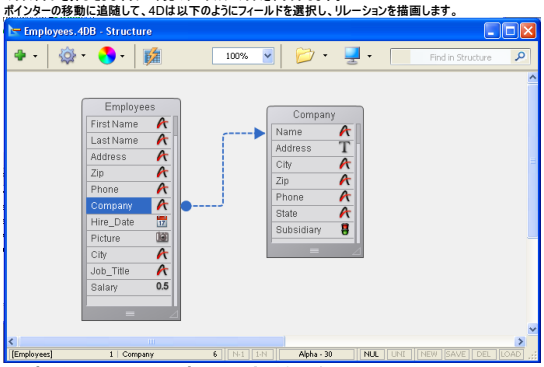
ストラクチャーエディターあるいは4DのSQLコマンドを使用して、リレーションを作成・削除できます。この節ではデザインモードで手作業でリレーションを作成する方法を説明します。4DでSQL文を使用する点についての詳細は、4D SQLリファレンスマニュアルを参照してください。

### リレーションの作成

2テーブル間のリレーションを手作業で作成する場合、2つのフィールド間で線を引くか、ツールバーのオブジェクト追加メニューを使用します。

ストラクチャーエディターでリレーションを作成するには:

1. ストラクチャーエディターウィンドウで、リレーションの外部キーとなるフィールドにカーソルを重ねます。
2. マウスボタンを押したまま、リレート先のテーブルにマウスをドラッグします。
   ポインターの移動に追随して、4Dは以下のようにフィールドを選択し、リレーションを描画します。

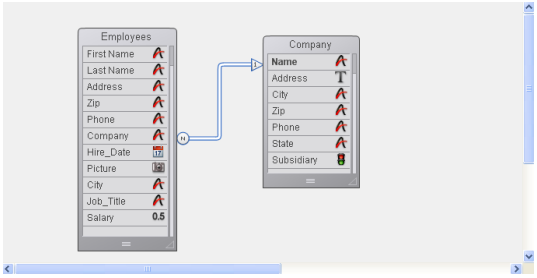



3. 1テーブルの主キーフィールドまでドラッグして、マウスボタンを離します。

オブジェクト追加メニューを使用してリレーションを作成するには:

1. それぞれ異なる2つのテーブルに属する同じタイプのフィールドを2つ選択します。
   選択の順番により、リレーションの方向が決定されます。最初に選択されたフィールドが外部キー (Nフィールド) で、2番目に選択されたフィールドが主キー (1フィールド) です。1フィールド側を選択する際はCommand (Mac OS) またはCtrl (Windows) を押しながらクリックします。
2. ストラクチャーエディターツールバーのオブジェクト追加メニューで**リレーション**オプションを選択します。

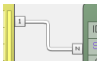
4Dはストラクチャーエディターの2テーブル間にリレーションを作成し、矢印で描画します。コネクターの始点および終点の形状、およびNや1の文字でリレーションの方向を判断できます:




インスペクターパレットにはリレーションのプロパティが表示されます (リレーションプロパティ参照)。

#### SQLで作成されるリレーション

SQLを使用して作成されたリレーションは異なる表示 (四角のコネクター) が使用されます:



### リレーションの削除

4Dではリレーションをいつでも削除できます。削除を行うには:

1. 削除するリレーションを選択します。
   リレーションが選択されると、外周が青で表示されます。
2. リレーション上で右クリックし、コンテキストメニューから**削除**コマンドを選択します。
   または
   **Delete**または**Backspace**キーを押します。

両ケースで警告ダイアログが表示されるので、処理を続行するかキャンセルするかを選択できます。処理を続行すると、4Dはリレーションの矢印を削除し、テーブルのリレーションは解除されます。Nフィールドや1フィールドのインデックスはそのまま残されます。

### リレーションの再設定

リレーションはいつでも再設定できます。例えば誤ったフィールド間でリレーションを設定してしまったり、リレーションのプロパティを変更したい場合に、これを行うことになるでしょう。4Dではリレーションの線を再び引くことで、リレーションを再設定できます。

複雑なストラクチャーの場合、リレーションのコンテキストメニュー内にある**Nフィールド選択**や**1フィールド選択**コマンドを使用して、リレーション元フィールドとリレート先フィールドを選択することができます。

同じ2つのフィールド間でリレーションを再設定するには、ストラクチャーエディター上でリレーションをダブルクリックします。4Dはインスペクターパレットにリレーションプロパティを表示します。ここで必要な変更を行うことができます。
リレーションを再設定して1テーブルのフィールドを変更するには、Nフィールドからリレーションを再描画します。
リレーションを再設定してNテーブルのフィールドを変更するには、まずリレーションを削除して、新しいリレーションを作成します。

## リレーションのタイプ

最も一般的なリレーションはNテーブルと1テーブル間で使用されるもので、N対1リレーションと呼ばれます。しかしN対Nや1対1リレーションを作成することもできます。すべてのリレーションはマニュアルあるいは自動リレーションにできます。

**自動およびマニュアルリレーション**

リレーションは自動またはマニュアルで使用できます。

自動リレーションでは、リレートしたテーブルのレコードがカレントになると、4Dは対応するレコードをカレントセレクションにします。カレントセレクションになった一連のレコードは表示、印刷、更新、そして検索や並び替えの対象として使用できます。プログラミングは必要ありません。

マニュアルリレーションでは、開発者が4Dに対し、対応するレコードをメモリにロードするかどうかを指示します。この制御を行うにはメソッドを使用します。リレートしたテーブルを制御するメソッドを作成するための情報は4Dランゲージリファレンスマニュアルを参照してください。

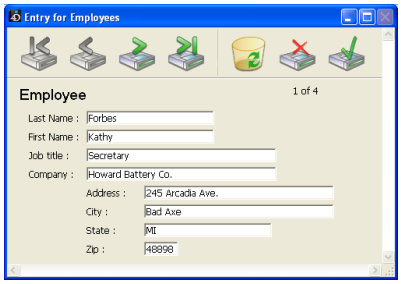
すべての対応するレコードを毎回ロードする必要がないようなアプリケーションでは、パフォーマンスを最適化するためにマニュアルリレーションを選択するでしょう。例えば3つ以上のテーブルが互いにリレートしているようなストラクチャーでは、リレートしたレコードのメモリへのロードを制御したくなるでしょう。また2つのテーブル間で2つの異なるリレーションを行う際も、マニュアルリレーションを使用する必要があります。2つのテーブル間では1つの自動リレーションしか存在できません。対してマニュアルリレーションはいくつでも存在できます。

**N対1リレーション**

2つのテーブル間でリレーションを作成する場合、リレーションで主キーを含むテーブルを1テーブル、外部キーを含むテーブルをNテーブルと呼びます。1テーブル中の1レコードがNテーブル中のNレコードにリレートし、Nテーブル中のNレコードが1テーブル中の1レコードにリレートするため、それぞれ1テーブル、Nテーブルと呼ばれます。このタイプのリレーションは*N対1*リレーションと呼ばれます。

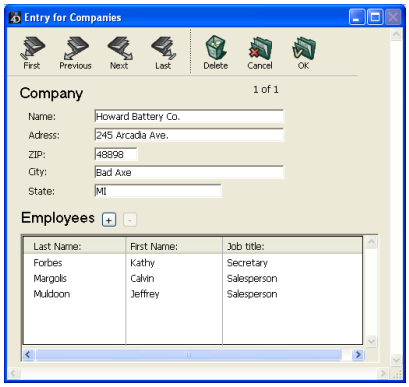
従業員と会社間のリレーションにおいて、[Companies]テーブルは1テーブルであり、[Employees]テーブルはNテーブルです。1つの会社レコードは複数の従業員レコード (例: 会社に勤めるすべての従業員) にリレートし、複数の従業員が1つの会社 (例: 彼らが働く会社) にリレートします。例えば[Companies]テーブルにAcmeレコードが1つだけあり、Acmeに勤務する従業員のレコードが複数[Employees]テーブルに登録されることがあります。

[Employees]テーブル中のいずれかのレコードがカレントになると、4Dは[Companies]テーブルから対応するレコードを1つロードします。[Companies]テーブル中のフィールドがフォーム上に配置されていれば、その値が自動で表示されます。
下図では、[Employees]テーブルのレコードが1つロードされたときに[Companies]テーブルのレコードが特定され、[Employees]テーブルレコードのフォーム上に会社の所在地と電話番号が表示されている様子を示しています:



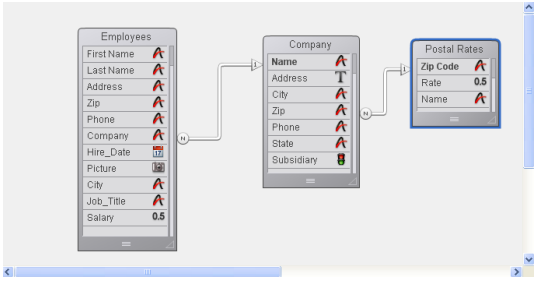

対して[Companies]テーブルのレコードが1つカレントになると、4Dは[Employees]テーブルレコードのセレクションを作成し、フォーム上に表示します。このリレーションは他のテーブルの複数のレコードを指定するため、複数の従業員の名前や役職が表示されます。フォームに表示されているレコードだけがメモリにロードされます。

下図では、[Companies]テーブルのレコードが1つロードされたときに複数の[Employees]テーブルのレコードが特定され、[Companies]テーブルレコードのフォーム上に、その会社で働く従業員の名前と役職がリスト表示されている様子を示しています:




1テーブルとNテーブルの区別は、リレーションごとに異なります。あるテーブルはあるリレーションにおいては1テーブルとなり、別なリレーションではNテーブルとなることがあります。リレーションにおいてテーブルは1つだけしか主キーをもつことができませんが、外部キーは複数持つことができます。

例えば[Employees]テーブル中の全員に製品サンプルを送付するとしましょう。[Postal Rates]テーブルを作成し、郵便番号と、郵便番号ごとの配送料を登録します。このストラクチャーを使用すると、従業員ごとに送付のための料金を含む住所ラベルを印刷することができます。下図はデータベーストラクチャーに追加された[Postal Rates]テーブルを示しています。




[Postal Rates]テーブルのZip Codeフィールドは主キーであり、[Postal Rates]テーブルは1テーブルです。[Companies] テーブルのZipフィールドはリレーションの外部キーフィールドです。Zipフィールドは外部キーなので、重複した値を登録することができます。つまり[Companies]テーブルは[Postal Rates]テーブルに対するNテーブルです。

そんなわけで、あるテーブルが1テーブルかNテーブルかは他のテーブルに対するリレーションで決定されます。[Companies]テーブルは[Postal Rates]テーブルとのリレーションではNテーブルであり、[Employees] テーブルとのリレーションでは1テーブルです。

**1対1リレーション**

1対1リレーションは特別なケースで使用されます。通常1対1リレーションはひとつのテーブルに合成できます。

以下は1対1リレーションを使用するいくつかの理由です:

- あまりにも多くのフィールドをテーブルに定義する必要があるため、論理的にそれらをグループ分けしたほうがデータベースの動作が速くなり、管理も容易になるケース。
- 特定のフィールドへのアクセスを制限したいケース。分離したテーブルを使用すれば、それぞれのテーブルごとにことなくアクセス権を割り当てることができます。

**N対Nリレーション**

時にはあるテーブルのNレコードと他のテーブルのNレコードをリレートさせる必要があります。これはN対N (あるいは多対多) リレーションと呼ばれます。

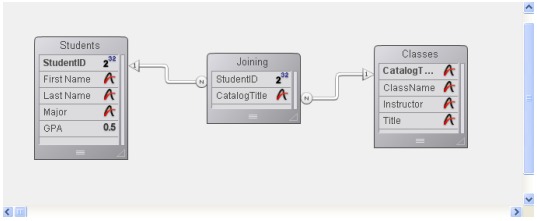
N対Nリレーションの例に、受講履歴のデータベースがあります。データベースに2つのテーブル[Students]と[Classes]があるとします。ひとりの生徒は複数の講座を受講し、ひとつの講座には複数の生徒が在籍します。ひとりの生徒が受講するすべての講座を見る必要があり、ある講座を受講するすべての生徒を見る必要もあります。

N対Nリレーションの例はほかにもあります:

- [Suppliers]と[Products]: 問屋は多くの製品を扱い、各製品は複数の問屋から提供されます。
- [Employees]と[Account]: 各従業員は複数の取引を扱い、各取引は複数の従業員により処理されます。
- [Movies]と[Actors]: 各映画には複数の出演者がいて、各出演者は複数の映画に出演します。

4Dを使用して自動N対Nリレーションを作成できます。キーは関連する2つのテーブルとそれぞれ1対Nでリレートする中間テーブルを作成することにあります。そのうえで、すべての必要なレコード追跡と表示を行うリストフォームと詳細フォームを作成することができます。この節ではN対Nリレーションを処理するために自動リレーションを使用する方法を説明します。

下図は生徒の受講状態を管理するためのデータベースに定義された3つのテーブル[Students], [Classes], [Joining]を示しています。このデータベースストラクチャーはこの節でN対Nリレーションを説明するために通して使用されます。




[Students]テーブルは1テーブルで、各生徒ごとに1つのレコードが作成され、名前、専攻、GPA (成績平均点) が格納されます。StudentIDフィールドにより生徒はユニークに識別されます。
[Classes]テーブルも1テーブルです。各講座ごとに1つのレコードが作成され、講座名と講師名が格納されます。CatalogTitleフィールドにより、講座がユニークに識別されます。
中間テーブルである[Joining]は他の2つのテーブルに対してNテーブルとなります。このテーブルにはN件の学生とN件の講座レコードが登録されます。このテーブルのフォームは他の両テーブルのデータを入力するため、および他の両テーブルの情報を表示するために使用されます。
3つのテーブルを使用することで、データが効率的に格納されます。生徒の完全なデータは1つのレコードのみに格納され、講座の完全なデータも1つのレコードにのみ格納されます。生徒の各受講データは一度だけ格納されますが、結果すべての組み合わせを表現することができます。

#### N対Nリレーションのデータを入力する

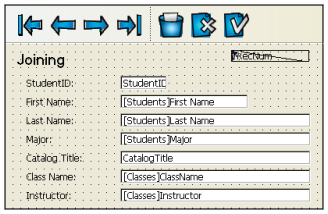
中間テーブル (この例題では[Joining]テーブル) を使用して、他の両テーブルからの情報を入力および表示します。[Joining]テーブルに入力できる各レコードはリレートしている他のテーブル (生徒と講座) です。[Joining]テーブルのレコードにはリレーションを実行するための2つの情報、studentIDとcatalogTitleだけが格納されます。以下は[Joining]テーブルに作成された新しいレコードの例です。

このレコードはJeffrey T. SpauldingがJournalism講座を受講していることを示しています。このレコードは実際には2つのテーブルからの情報を表示しているにすぎません。

同様のレコードが、各生徒が受講するすべての講座に対して作成されます。実際にはStudentIDとCatalogTitleだけが[Joining]テーブルに格納されます。各レコードには特定の生徒が受講する特定の講座が保存されます

**注:** [Joining]テーブルのレコードがロードされると、リレートしたテーブルのレコードセレクションが自動で作成されます。この場合セレクションは対応する生徒と講座レコードで構成されます。ここで他のテーブルのセレクションを表示すると、1つのレコードだけが表示されます。すべてのレコードを表示するには**クエリ**メニューから**すべてを表示**を選択します。

このレコードの入力フォームは以下のようになります。フォーム中に[Students]および[Classes]テーブルのフィールドが含まれている点に留意してください。

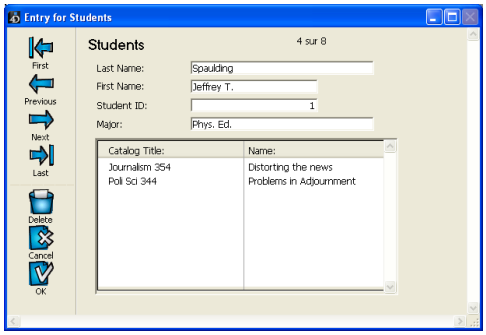



データはStudentIDとCatalogTitleフィールドにのみ入力されます。StudentIDが入力されると、4Dはリレートした[Students]テーブルから対応する情報を探し、それをLast Name、First Name、Majorフィールドに表示します。同様にCatalogTitleが入力されると4Dは講座を[Classes]テーブル内で検索し、その情報を表示します。

#### 情報をサブフォームに表示する

これら3つのテーブルの情報をサブフォームを使用して表示することができます。生徒のレコードでは生徒が受講するすべての講座のリストを表示でき、講座レコードではその講座を受講するすべての生徒のリストを表示できます。

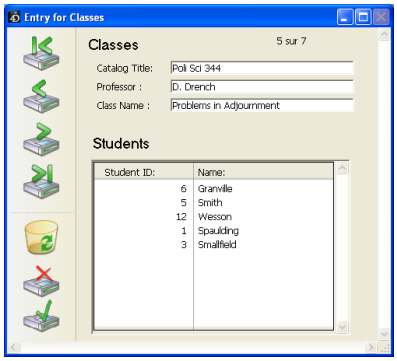
生徒のレコードに講座のリストを表示するには、サブフォームを使用します。サブフォームの作成に関する情報はサブフォームの作成と設定を参照してください。




上図は[Students]テーブルのレコードです。レコードの上部には生徒に関する情報が表示されています。この生徒が受講している2つの講座の情報が下部に表示されていますが、これは受講情報が格納された[Joining]テーブルのデータが使用されています。

サブフォームは[Joining]テーブルのものであり、[Classes]テーブルではない点に留意してください。[Joining]テーブルは生徒のレコードから講座のレコードにリレートする情報を格納しています。サブフォームには[Classes]テーブルのClassNameフィールドが置かれています。[Joining]と[Classes]テーブル間のリレーションのために、4Dは自動で正しい講座名を表示します。

以下はある講座とその講座を受講する生徒を表示するレコードです:




これは[Classes]テーブルのレコードで、講座の情報と、それを受講する生徒のリストが表示されています。生徒の情報もまた[Joining]テーブルから表示されます。

このサブフォームの例題で、表示されたすべてのフィールドにデータを入力することができます。例えば講座レコードに新しい学生を入力するには、サブフォームに表示された最後の生徒レコードに移動し、**Ctrl+Shift+/** (Windows) または **Command+Shift+/** (Mac OS) を押して新しいレコードを作成します (ショートカットはデータベース設定で変更できます、を参照)。適切なStudentIDを入力すると、レコードの他の情報が表示されます。

**データベースリレーションの解析**

データベースに設定されたリレーションはデータベースの動作において重要な役割を持ち、テーブル間の情報フローを制御します。

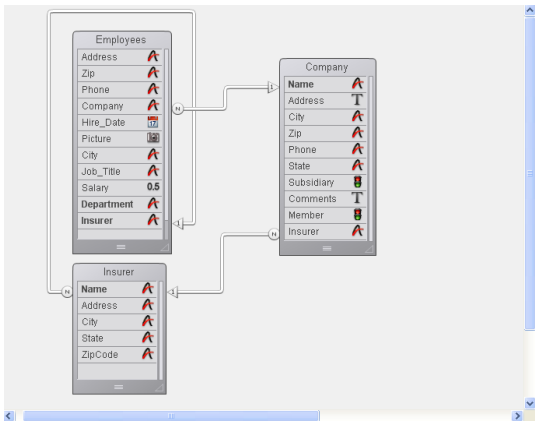
自動リレーションが設定されたレコードが詳細フォームを使用してディスクからロードされると、リレートしたテーブルの対応するレコードがカレントセレクションになります。リレーションが1つだけレコードを選択した場合、そのレコードがディスクからロードされます。リレーションにより複数のレコードが選択されると、レコードの新しいカレントセレクションがそのテーブルに作成され、カレントセレクションの先頭レコードがディスクからロードされます。ディスクからロードされたレコードはそのテーブルのカレントレコードと呼ばれます。

この節の例題では、リレーションは3つのテーブル間だけで設定されていました。実際のアプリケーションでは、リレーションは複数のテーブル間で作成され、チェーンのように次々とアクティブ化されます。リレーションがアクティブ化されるたびに、4Dはリレートしたテーブルのレコードセレクションを作成し、レコードをディスクからロードします。ディスクからロードされたレコードはそのテーブルのカレントレコードとなり、そのテーブルがさらに自動リレーションを持てば、4Dは次のリレートしたテーブルのセレクションを作成しカレントレコードをロードします。

テーブルリレーションが正しく設定されていないと、テーブル間の情報の流れは混乱し、壊れてしまいます。注意すべきリレーション構造を以下に紹介します。

**循環リレーション**

循環リレーションは情報が無限にループするように設定されたリレーションです。下図は循環リレーションの例であり、[Employees]テーブルが[Company]テーブルにリレートし、[Company]テーブルは[Insurer]テーブルにリレートし、[Insurer]テーブルは[Employee]テーブルに戻ってリレートしています。



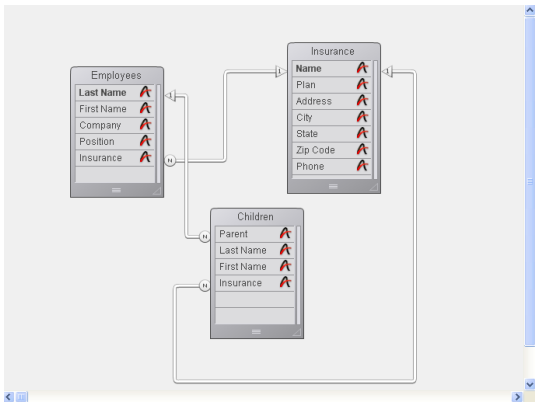

[Employee]テーブルのレコードがディスクからロードされると、4Dはリレートする[Company]テーブルのレコードをロードします。このレコードは[Company]テーブルのカレントレコードとなります。これによりリレートする[Insurer]テーブルのレコードがロードされます。

もしテーブルリレーションがこのまま続行されると、[Insurer]テーブルにリレートする[Employee]テーブルのレコード (会社により保険が掛けられたすべての従業員) が選択され、先頭レコードがカレントレコードになります。このカレントレコードは、このリレーションチェーンが開始された元の従業員レコードとは異なるかもしれません。この状況では、4Dはどのレコードが実際のカレントレコード化を知るすべがありません。

4Dはこの種の循環リレーションを検知すると、チェーンの最後のテーブルでリレーションが停止します。この場合、[Insurer]と[Employee]テーブル間のリレーションは実行されません。

**同じテーブルに対する複数のリレーション**

カレントレコード間での同様の衝突が、同じテーブルに複数のリンクがある場合にも発生します。
1つのテーブルに対して一度に複数のカレントレコードを持つことはできないため、同じテーブルに対する複数のテーブルからの自動リレーションを管理することはできません。
以下のストラクチャーは同じテーブルにリレートするテーブルとそのサブテーブルを表しています。

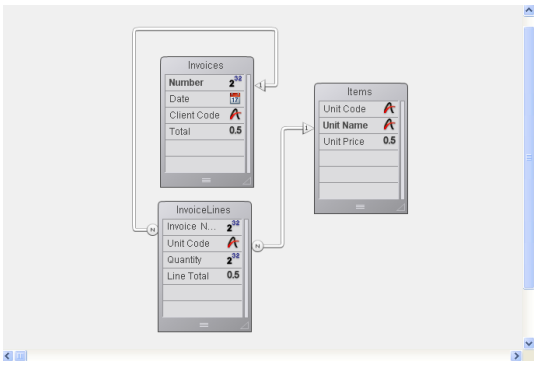



ユーザーが[Employees]テーブルで作業しているとき、[Insurance]テーブルのリレートしてレコードがロードされ、それがこのテーブルのカレントレコードとなります。
しかし[Children]と[Insurance]テーブル間にもリレーションがあります。つまり、[Children]テーブルのカレントセレクションの先頭レコードに基づく[Insurance]テーブルのリレートしたレコードもロードされることを意味します。親の保険会社と子の保険会社が異なる場合、このリレーション構造は問題となります。この場合4Dはリレーションの続行を停止しません。両方のリレーションが実行されますが同時ではありません。
この種のストラクチャーを使用したい場合はマニュアルテーブルリレーションを使用し、4Dランゲージリファレンスマニュアルで説明されているコマンドを使用してリレーションを制御しなければなりません。

自動リレーションでは管理できないリレーション構造のもう一つの例は、2つのテーブル間で複数のリレーションが設定されているケースです。1テーブルのリレートしたフィールドのデータを更新するたびに、他のテーブルのカレントレコードが変更されてしまします。この状況では自動リレーションを使用することはできません。

**複数レコードからのリレーション**

テーブルには1つだけカレントレコードが存在できるため、セレクション中のレコードすべてにリレーションが実行されるわけではありません。以下のストr買う茶ーのような請求書システムを考えてみましょう:




[Invoices]テーブルのレコードが使用されると、その請求書の請求明細である[InvoiceLines]テーブルのセレクションが作成されます。しかし[InvoiceLines]テーブルの先頭レコードに対応する[Items]テーブルのレコードだけが選択されます。[Items]テーブルのセレクションは請求書のすべての項目の情報を含むわけではありません。

しかし、[InvoiceLines]を[Invoices]テーブルのサブフォームとして配置した場合は、4Dは請求明細を呼び出すたびに項目へのリレーションを実行して、項目名を各行に表示させることができます。

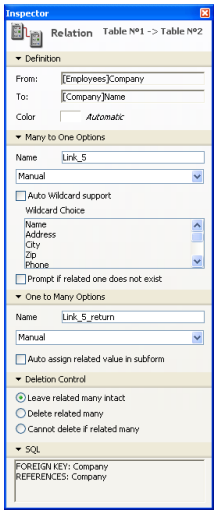
## リレーションプロパティ

リレーションは常にNテーブルから1テーブルに向けて描画します。

リレートされるフィールドは同じか互換性のある型でなければなりません。以下のフィールド型を主キーおよび外部キーフィールドとして使用できます:

- 文字列 (文字およびテキスト)
- 数値 (実数, 整数, 倍長整数, 64ビット整数, またはフロート)
- 時間
- ブール
- 日付

主キーには通常インデックスを設定します (必須ではありません)。リレーションプロパティは インスペクターパレット 、または特定の項目についてはリーレーションを右クリックして表示されるコンテキストメニューから設定できます。




### 定義

定義エリアには外部キーと主キーフィールドが表示されます。

- **開始**: "開始"フィールドにはこのリレーションのNテーブルの外部キーが表示されます。コンテキストメニュー中でこのフィールドはNフィールドと呼ばれます。
- **1フィールド**: "1フィールド"には1テーブルの主キーフィールドが表示されます。コンテキストメニュー中でもこのフィールドは"1フィールド"と呼ばれます。
  リレーションはNテーブルの外部キーから1テーブルの主キーに向かって描画します。
- **カラー**: このポップアップメニューを使用して、ストラクチャーエディターに表示されるリレーションの線のカラーを設定できます。リレーションのカラーはプロパティーとはリンクされていません。
  このカラーはリレーションのコンテキストメニューを使用して設定することもできます。

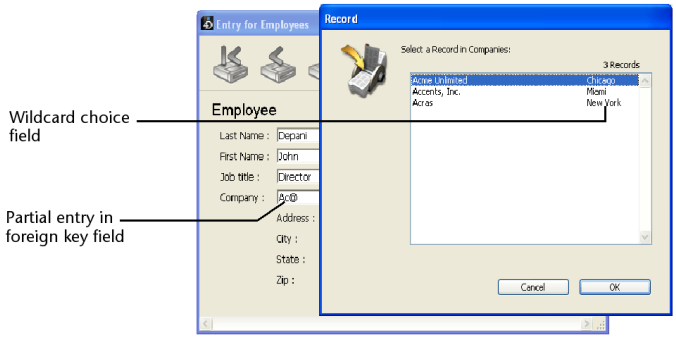
### N対1オプション

N対1オプションはNテーブルのレコードが開かれたときの動作に影響します。

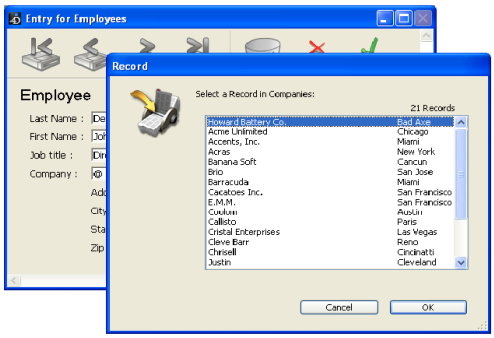
- **名前**: N対1リレーションの名前を入力するエリア (任意)。
- **手動/自動**: このメニューで、Nテーブルから1テーブルへリンクされたリレーションの動作モードを設定します。自動オプションを選択すると、自動機能が有効になります。例えばアプリケーションモードで[Employees]テーブルのレコードが開かれると、自動で対応する[Companies]テーブルのレコードも開かれます。この設定では従業員が働く会社の情報を4Dが自動で表示します。
  デフォルトでは**手動**オプションが選択されています。この場合、ランゲージを使用して開発者が1テーブルのレコードのロードとアンロードを管理しなくてはなりません。
  この点に関する詳細は自動およびマニュアルリレーションを参照してください。
  リレーションのコンテキストメニューを使用して自動1リレーションを設定することもできます。
- **自動ワイルドカードサポート**: このチェックボックスにチェックが入れられていると、ユーザーがタブキーを押したり、フィールド外をクリックしたときに、Nテーブルの外部キーに入力された値に自動で見えないワイルドカード文字 (@) が追加されます。ユーザーが値の一部を入力すると、4Dは1テーブルから一致する値を探します。4Dが1つだけレコードを見つけると、その値が自動で入力されます。4Dが複数のレコードを見つけると、それらがリスト表示され、ユーザーはその中から一つを選択できます。
- **リレートした1レコードがないとき知らせる**: このチェックボックスをチェックすると、リレートした1レコードが存在しない場合にそのレコードをユーザーが作成できるようにするダイアログボックスを、4Dが表示します。ここにチェックされていると、Nテーブルからリレートしたフィールドに値を入力すると、4Dはリレートした1テーブルに対応するレコードが存在するかどうかを検証します。一致するレコードを見つけることができない場合、以下のダイアログが表示されます:

  このダイアログボックスから、Nテーブルにデータを入力中に、対応する1テーブルのレコードを作成することができます。例えば[Invoices]と[Customers]テーブルからなる請求データベースがあるとします。[Customers]テーブルにまだ登録されていない顧客あての請求データを[Invoices]テーブルに登録すると、[Invoices]テーブルのレコードを受け入れる際に、4Dは対応するレコードを[Customers]テーブルに作成するか尋ねできます。
  **リレートした1レコードがないとき知らせる**のチェックを外すと、このダイアログボックスを表示させないようにできます。これはメソッドでリレートした1テーブルのレコードを作成するような場合に行います。
- **ワイルドカード選択リスト**
  ワイルドカードフィールドのリストを使用して、(ユーザーがデータ入力する際リレートしたフィールドにワイルドカード@を入力すると表示される) 値のリストに追加で表示されるフィールドを選択できます。普通、レコードを識別するために適切なフィールドを選択します。
  データ入力時にこの機能は以下のように動作します: 4Dはユーザーが Nテーブルの外部キーフィールドにデータを入力する際、1テーブルの値を参照できるようにします。ユーザーはリレートしたフィールドに標準のワイルドカード文字 (@) を使用します。そうすると4Dはリレートした1テーブル中で対応する入力を検索します。
  ワイルドカード文字は2つの方法で使用できます: 部分的な入力を補完する、または有効な値のリストを表示する。リストが表示されたら、ユーザーはリスト中から値を選択できます。追加のワイルドカード選択フィールドをリレートしたフィールドに表示できます。
  例えばユーザーが[Employees]テーブルにレコードを作成しているとします。会社名フィールドにAcme Unlimitedとタイプする代わりに、ユーザーはAc@とタイプし、タブキーを押して次のフィールドに移動できます。@は4Dのワイルドカード文字なので、この入力は"Acで始まる文字列"を意味します。4Dはリレートしたテーブルでこの入力に一致するレコードを探します。1つだけ見つかると、その データがフィールドに入力され、データ入力順に従い次のフィールドが選択されます。
  次の図はワイルドカードを使用することでどのように動作するかを示しています。

  4Dが複数のレコードを見つけると、見つかった値のリストが表示され、ユーザーはそのうち1つを選択できます。次の図はこのときに表示されるリストです:




選択する会社をユーザーが見つけやすくするために、リストに追加で表示するフィールドを選択できます。これがワイルドカード選択フィールドで、インスペクターパレットのリレーションプロパティで設定します。
上の図では会社名のほかに都市名も表示されています。このワイルドカード選択フィールドにより、ユーザーはAccents, Inc.またはAcme Unlimitedの中から、会社がChicagoにあることを思い出して、選択できます。
[Company]テーブルのすべての会社をリストするには、@だけを入力します。4Dはすべての会社をリスト表示し、ユーザーはその中から目的の会社を選択できます。次の図はすべての会社がリスト表示された様子です:

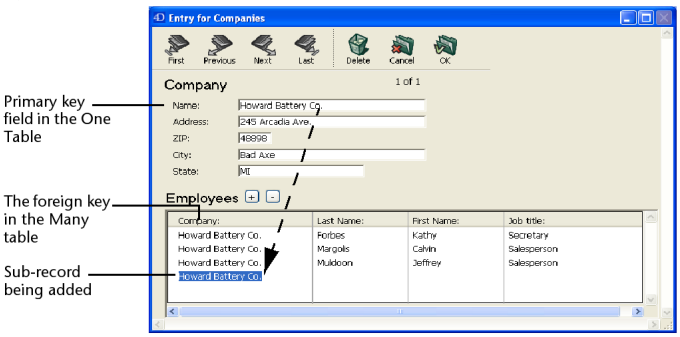



レコード選択ウィンドウはサイズ変更できます。

**1対Nオプション**

1対Nオプションは逆方向の自動リレーションを制御します。

- **名前**: 1対Nリレーションの名前を入力するエリア (任意)。
- **手動/自動**: このメニューで、1テーブルからNテーブルへリンクされたリレーションの動作モードを設定します。**自動**オプションを選択すると、自動機能が有効になります。例えばアプリケーションモードで[Company]テーブルのレコードが開かれると、自動で対応する[Employees]テーブルのレコードもロードされます。この設定では会社で働く従業員をサブフォームに4Dが自動で表示します。
デフォルトでは**手動**オプションが選択されています (自動動作なし)。この点に関する詳細は自動およびマニュアルリレーションを参照してください。
リレーションのコンテキストメニューを使用して自動1対Nリレーションを設定することもできます。
- **サブフォームにリレート値を自動代入する**: このオプションを使用すると、データ入力中に、Nテーブルの外部キーフィールドに1テーブルの主キーの値を自動で代入させられます。このオプションは1対Nリレーションで**自動**オプションが選択されている場合にのみ利用価値があります。
このオプションは、リレートしたNテーブルのサブフォームを持つ1テーブルの詳細フォームでデータを入力する際に影響があります (サブフォームについてはサブフォームとウィジェットを参照)。**サブフォームにリレート値を自動代入する**が選択されていると、ユーザーがサブフォームにレコードを追加する際に、リレート値がNテーブルの外部キーフィールドに自動で代入されます。
[Company]と[Employees]テーブルのリレーションにおいて、[Company]テーブルが1テーブルであり[Employees]はリレートしたNテーブルです。
各会社毎に[Company]テーブルに1レコードが作成され、[Employees]テーブルにはその従業員レコードが複数あります。Nテーブルの外部キーフィールドがサブフォームに表示されていれば、このオプションの効果を視認することができます。サブフォーム中でサブレコードが追加されるたびに、自動でフィールドの値がコピーされます。




このオプションはNテーブルの外部キーフィールドがサブフォームに表示されていなくても動作します。主キーフィールドの値は"内部的に"自動で外部キーフィールドにコピーされます。
**サブフォームにリレート値を自動代入する**にチェックが入れられていない場合、作成されたサブレコードは自動では適切にリレートされません。
この場合は以下の方法でサブレコードをリンクしなければなりません:
  - 手作業でリレートのための値を入力する。先の例題においては作成したサブレコードごとに、会社名フィールドに"Howard Battery Co."を入力します。
  - またはサブレコードを作成する際に
    ```
    [TableN]Field1:=[Table1]Field1
    ```
    のようなコードを実行します。

**注:** Nテーブルのリレートレコードを作成した後に1テーブルの主キー値を変更するような場合、**サブフォームにリレート値を自動代入する**は効果がありません。このようなケースでは手作業あるいはコードを使用して外部キーの値も変更しなければなりません。

**削除制御**

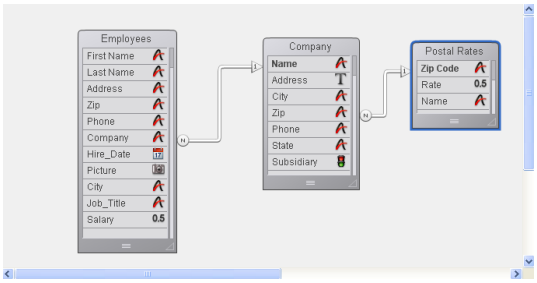
削除制御オプションでは、1テーブルのレコードが削除される際のNテーブルのリレートしたレコード削除のルールを指定します。通常ユーザーはカレントテーブルでないテーブルのレコードを削除することはできません。これは例えば[Employees]テーブルのレコードを削除するためには、まずこのテーブルをカレントテーブルにする必要があることを意味します。デザインモードではテーブルリスト中で目的のテーブルを選択することで、カレントテーブルにできます。

- **リレートしたNレコードを残す**: このラジオボタンを選択すると、ユーザーが1テーブルのレコードを削除しても、Nテーブルのレコードはそのまま残されます。結果Nテーブルには1テーブルに対応するレコードが無いレコードが残されることになります。この場合、Nテーブルのレコードがフォームにロードされても、1テーブルから対応するレコードはロードされません。

- **リレートしたNレコードを削除**: このラジオボタンを選択すると、ユーザーが1テーブルのレコードを削除すると、4Dは自動でリレートするNテーブルのレコードをすべて削除します。このプロパティを選択すると、1テーブルのレコードが削除されるときにNテーブル中に迷子のレコードが発生することを避けることができます。
- **リレートしたNレコードがあれば削除しない**: このラジオボタンを選択すると、Nテーブルにリレートしたレコードがある場合、1テーブルのレコードの削除を禁止します。このプロパティにより、誤ってレコードが削除されることを防ぐことができます。Nテーブルのレコードは、この選択に関わらず削除ができる点に留意してください。

**リレートしたNレコードを削除**と**リレートしたNレコードがあれば削除しない**ラジオボタンはデータベース理論で*参照整合性*と呼ばれるものを強化します。参照整合性が有効であるとき、4Dはリレートされたテーブル中の各レコードが常に1テーブルの1つのレコードに関連付けられていることを確信することができます。

複数のリレートしたテーブルがある場合、削除制御は各リレーションごとに連鎖して適用されます。例えば以下のようなストラクチャーで考えます。リレートしたNレコードを削除がそれぞれのリレーションで選択されているとき、[Postal Rates]テーブル (1テーブル) からレコードが削除されると、4Dはまず[Company]テーブルから関連するレコードを削除し、次に削除された会社に属する従業員レコードを[Employees]テーブルから削除します。




**削除制御設定に矛盾する状態にが発生した場合、4Dは削除を許可しません。例えば**[Company]テーブルと[Postal Rates]テーブル間のリレーションに**リレートしたNレコードを削除**が設定されていて、[Employees]テーブルと[Company]テーブル間には**リレートしたNレコードがあれば削除しない**が設定されている場合、削除は行われず、[Company]と[Employees]テーブルに変更はありません。

**SQL**

インスペクターパレットのSQLエリアにはSQLランゲージからストラクチャーを操作する際に有用な情報が表示されます。
リレーションの場合エリアにはFOREIGN KEYとREFERENCESプロパティが表示されます。

## ストラクチャー定義の書き出しと読み込み

4Dではデータベースストラクチャー定義をXMLやHTMLファイルとして書き出すことができます。またXMLフォーマットで保存されたストラクチャー定義を使用して、新しい4Dデータベースを作成することもできます。この新しい機能により、いくつかの可能性が生まれます:

- ストラクチャーをカスタマイズされたフォーマット (レポートや表など) で見せたり、他のツールで解析することができます。
- 定義ファイルからデータベースを生成することができます。

### 4Dストラクチャー定義のフォーマット

4Dストラクチャー定義はXMLフォーマットに基づきます。ストラクチャー定義はテキストエディターで開くことができます。またXMLフォーマットのデータは、特にXSL変換などを使用することで、他の形式に変換することが可能です。さらに4Dは.XSLファイルを使用してストラクチャー定義をHTMLフォーマットで書き出すことができます。

ストラクチャー定義にはテーブル、フィールド、インデックス、リレーション、それらの属性、およびストラクチャー定義を完全なものにするための様々なデータが含まれます。4Dストラクチャー定義の内部的な文法はDTDファイルに定義されています。このファイルはXMLファイルの妥当性検証にも使用されます。
4Dが使用するDTDファイルは4Dアプリケーション内のDTDフォルダーにまとめられています。ストラクチャー定義には**base_core.dtd**と**common.dtd**ファイルが使用されます。4Dストラクチャー定義に関する詳細は、これらのファイルおよびそこに書かれたコメントを参照してください。

### ストラクチャー定義の書き出し

4Dではストラクチャー定義をXMLおよびHTMLフォーマットで書き出すことができます。必要に応じてフォーマットを選択してください:

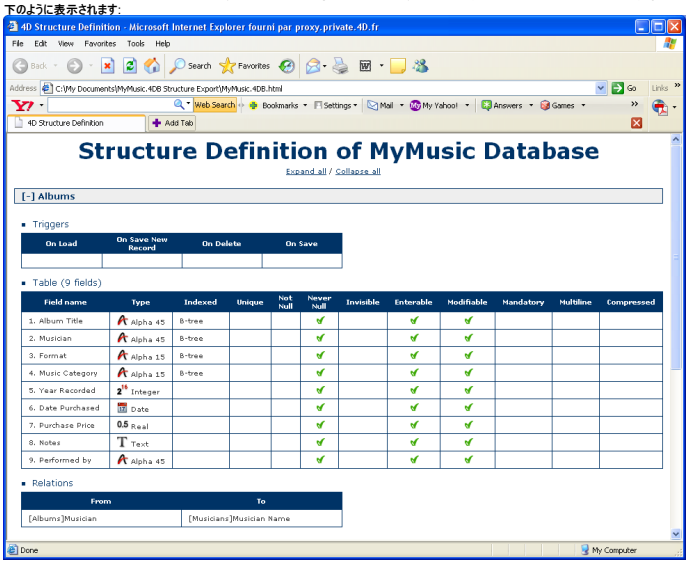
- **XMLフォーマット**: XMLフォーマットのストラクチャー定義はテキストエディターを使用して開いたり、(カスタムXSL変換や他の解析ツールで読み込むなどして) 使用することができます。
  新しいデータベースをストラクチャー定義から作成したい場合もこのフォーマットを選択してください。
- **HTMLフォーマット**: このフォーマットを使用すると、Webブラウザーを使用してストラクチャー定義を表示したり印刷したりできます。

XMLフォーマットでストラクチャー定義を書き出すには:

1. 4Dの**ファイル**メニューから**書き出し > ストラクチャー定義をXMLファイルに書き出し...**を選択します。
   標準のファイルを保存ダイアログボックスが表示されます。
2. 書き出すファイルの名前と保存場所を指定してダイアログを受け入れます。

HTMLフォーマットでストラクチャー定義を書き出すには:

1. 4Dの**ファイル**メニューから**書き出し > ストラクチャー定義をHTMLに書き出し...**を選択します。
   HTMLファイルの保存先フォルダーを指定するダイアログが表示されます。
2. **新規フォルダー**ボタンをクリックするか、既存のフォルダーを選択します。
   4Dは指定された場所に自動で"*Mystructure.4db* Structure Export"フォルダーを作成し、ストラクチャー定義をそのフォルダー内に書き出します (*Mystructure.4db*は実際のデータベース名になります)。
   デフォルトのブラウザーで書き出されたファイルを表示するかどうかを尋ねるダイアログが表示されます。HTMLフォーマットのストラクチャー定義は以下のように表示されます:




#### XSL変換のカスタマイズ

HTMLフォーマットのストラクチャー定義を生成するために、4Dはアプリケーションの/Resource/*language*.lprojサブフォルダーに配置された"Structure_to_html.xsl"ファイルをデフォルトで使用します。

**注:** このファイルが所定の場所に存在しない場合、書き出しダイアログからHTMLでの書き出しを行うことはできません。

カスタムXSLスタイルシートファイルを使用することで、この変換をカスタマイズできます。これを行うには、"Structure_to_html.xsl"という名前のファイルを作成し、.4dbファイルと同階層におきます (デフォルトファイルを複製できます)。HTMLフォーマットのストラクチャー定義を生成する際に4Dはこのファイルを使用します。

### ストラクチャー定義からデータベースを作成する

XMLフォーマットで書き出されたストラクチャー定義を使用して、新規に同じデータストラクチャーを作成することができます。この場合ストラクチャー定義は、必要に応じて複製が可能なストラクチャーテンプレートと考えることができます。

XMLストラクチャー定義はそのように使用することができるほか、XMLエディターで事前に修正を加えることもできます。プログラミングによってストラクチャーを生成するために使用されるすべてのタイプのメカニズムが対象となります。

さらに、4Dストラクチャー定義XMLファイルの内部フォーマットは公開されています (前述の"4Dストラクチャー定義のフォーマット"参照)。他のデータベース環境やデザインアプリケーションを使用してこのタイプのファイルを生成し、4Dデータベースを自動で作成することができます。

ストラクチャー定義からデータベースを作成するには:

1. 4Dの**ファイル**メニューから**新規 > ストラクチャー定義を使用したデータベース...**コマンドを選択します。
   標準のファイルを開くダイアログボックスが表示され、開く定義ファイルを指定できます。4Dストラクチャ定義の文法に準拠したXMLフォーマットのファイルを開かなければなりません (4DはDTDを使用してファイルを検証します)。
2. ストラクチャー定義XMLファイルを選択して**OK**をクリックします。
   4Dは作成するデータベースの名前と場所を指定するダイアログボックスを表示します。
3. データベースの名前と場所を指定して**保存**をクリックします。
   XMLファイルが妥当であれば、4Dは (開かれていれば) カレントデータベースを閉じ、ストラクチャー定義に基づく新しいストラクチャーを作成し、エクスプローラーウィンドウを表示します。デフォルトで空のデータファイルも作成されます。

# フォームの管理


概要

フォームウィザードを使用してフォームを作成する

空のフォームを作成する

フォームを編集する

フォーム名を変更する

詳細/一覧フォームを指定する

フォームを削除する

フォームを印刷する

フォームを保存する

## 概要

フォームはデータの入力、修正、印刷を行うためのインターフェースとなります。ユーザーは、フォームを使用してデータベースのデータをやり取りし、フォームを使用してレポートを印刷します。カスタムアプリケーションにおいては、フォームを使用してカスタムダイアログボックスやパレットを作成します。

通常、データベースの各テーブルには少なくとも2 つのフォームが存在します。1 つは画面上にレコードを一覧するフォームであり、もう1 つは一度に1 件のレコードを表示しデータの入力や修正に使用するフォームです。レコードを一覧するフォームは出力フォームまたはリストフォームと呼ばれ、一度に1 件のレコードを表示するフォームは入力フォームまたは詳細フォームと呼ばれます。リストフォームを使用してレコードを表示しているとき、任意のレコードをダブルクリックすると詳細フォーム上にそのレコードが表示されます。

### フォームのコンテンツ

カスタムアプリケーションでは、ランゲージを用いて使用するフォームを指定できます。例えば使用するスクリーンのタイプに応じてフォームを切り替える必要があるかもしれません。またランゲージを使用すれば、データベースの使用者ごとにフォームを切り替えることもできます。カスタムアプリケーションを作成する際は、カスタムダイアログボックスやフローティングパレットとして使用するフォームを作成できます。また複数のプロセスで同時に複数のフォームを使用するようプログラムすることもできます。

フォーム上には複数のテーブルのフィールドを配置し、表示できます。リレートした1テーブルのフィールドをフォーム上に配置し、直接リレートした1 テーブルに値を入力することが可能です。またサブフォームを組み込み、リレートしたNテーブルのレコード一覧を表示することもできます。サブフォームには別のテーブルのレコード一覧を表示できます。サブフォームを使用すると、ユーザーは別のテーブルのレコードの表示、入力、変更を行えます。例えば請求書アプリケーションでは、請求書の入力フォーム上でサブフォームを使用し、このサブフォーム上で請求書の明細行を入力できます。請求項目は請求書フォーム上に表示されますが、実際のデータはリレートしたNテーブルに格納されます。

データ入力に用いるフォームには、複数のサブフォームを配置できます。例えばコンタクト管理データベースの場合、電話番号用のサブフォームと予定用のサブフォーム、そしてその人との前回のコンタクト用のサブフォームを配置できます。各サブフォームにはそれぞれ異なるNテーブルのレコードが表示されます。

フォームにはテーブル中のフィールドの一部、またはすべてのフィールドを配置できます。例えば従業員用と管理者用、二種類の入力フォームを用意し、管理者にはより多くのフィールドへのアクセスを提供できます。また別のフィールドグループを画面表示用に使用し、さらにレポート印刷用に4つめのグループを使用することも可能です。
データベースへデータを登録したかどうかに関わらず、フォームはいつでも変更可能です。フォームを変更しても、ディスク上に保存されたデータへの影響はありません。

各フォームには1 つ以上の表示ページがあり、それぞれにフィールドやその他の入力可オブジェクトが表示されます。フィールドが1 ページ内に納まらない場合、ページを追加することができます。マルチページフォームを作成する際、ページ間を移動するためのボタンやタブコントロールを追加します。

また各フォームには背景ページ(ページ0)があり、すべての表示ページに表示されるオブジェクトをこのページに配置します。この背景ページを使用し、ページの"外観"を定義する背景画像やボタン、タブコントロール、その他のグラフィックオブジェクトなどを配置します。

**注:** 複数ページフォームを出力フォーム(例えば印刷用)として使用すると、最初のページだけが表示されます。

### フォームを作成する

4Dでは素早く標準のフォームを作成することができます。また、高性能なツールが用意されているため、洗練されたインタフェースを備えたフォームを作成するためのパワフルなツールも提供されます。データベースのニーズに適合したフォームを作成することができます。ポイント&クリック操作だけで、フィールドやボタン、変数などが配置された基本的なフォームを作成できます。

4Dにはフォームの作成や修正に使用する**フォームウィザード**と**フォームエディタ**、2つのツールがあります。

#### フォームウィザード

フォームウィザードを使用して、あらゆるタイプのフォームの作成を開始できます。フォームウィザードではリストから目的のフィールドを選択し、ドロップダウンリストから目的のフォームテンプレートを選択して、新規フォームを作成することができます。フォームテンプレートは、フォームの外観を決定します。テンプレートはフォームサイズやプラットフォームインタフェース、フォント属性、ボタンなどの特性を指定します。詳細はフォームウィザードを使用してフォームを作成するを参照してください。

#### フォームエディター

フォームエディターはオブジェクト指向の描画環境であり、フォーム上のオブジェクトを直接操作して、フォームをカスタマイズすることができます。例えばオブジェクトの位置変更やフォームウィザードではサポートされないオブジェクトの追加、タブコントロールを用いた複数ページフォームの作成、データ入力制約の指定によるビジネスルールの適用、フォームアクセス権の設定、フォームへのカスタムメニューの割り当て、フォーム使用時に自動実行されるフォームメソッドやオブジェクトメソッドの作成を行うことができます。詳細はフォームを編集するを参照してください。

### テーブルフォームとプロジェクトフォーム

4Dでは2つのカテゴリーのフォーム (**テーブルフォーム**と**プロジェクトフォーム**) を作成できます。基本的にテーブルフォームは特定のテーブルに属していて、そのためにデータベースに基づくアプリケーションの開発時に便利な自動機能の恩恵を得ることができます。プロジェクトフォームはテーブルに属さない独立したフォームです。このタイプのフォームは特にインターフェースダイアログボックスやコンポーネントを作成するために使用されます。

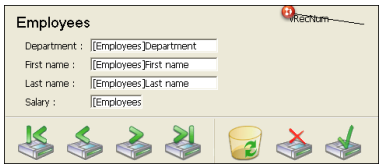
プロジェクトフォームを使用してより簡単にOS標準に準拠するインターフェースを作成できます。特にレコード一覧を表示する場合、(DIALOGコマンドを使用して) サブフォームにレコードセレクションを表示するプロジェクトフォームを呼び出すことが推奨されます。少々の追加プログラムを行うことで、MODIFY SELECTIONやDISPLAY SELECTIONコマンドよりも制御のしやすいフォームを作成できます。

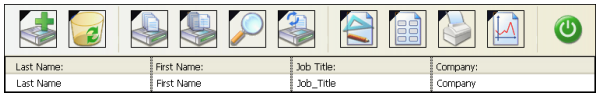
テーブルフォームとプロジェクトフォームはエクスプローラーのフォームページで別々にグループ化されます。

#### テーブルフォームの特徴

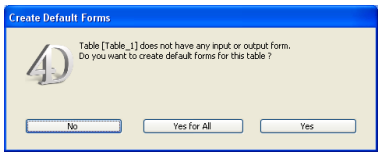
特定のテーブルに属するフォームです。各テーブルには、データをフィールドに入力したり画面上に表示するために、少なくとも1 つのフォームが存在すべきです。通常テーブルには入力フォームと出力フォームが別々に存在します。入力フォームはデータ入力に使用されます。このフォームには1 画面につき1 件のレコードが表示され、通常はレコードの保存や変更内容のキャンセルを行うためのボタン、レコード間を移動する(先頭レコード、最終レコード、前レコード、次レコード)ボタンが置かれます。出力フォームは1 行に1 レコードを納めたレコード一覧を表示します。クエリの結果は出力フォームに表示され、ユーザは出力フォーム上の行をダブルクリックして、入力フォーム上にそのレコードを表示することができます。

次の図は一般的な入力フォームと出力フォームを示しています:




テーブルのフォームを作成せずにレコードの表示を行おうとすると、4Dはデフォルトの入力フォームと出力フォームを作成するかどうかを尋ねてきます (これを行わないように設定することもできます)。




**注:** 環境設定の**自動フォーム作成**オプションを使用すると、例えば自動的にデフォルトフォームを作成するように設定することができます。この場合デフォルトフォーム作成ダイアログボックスは表示されません。詳細は一般ページを参照してください。

**はい** (または**すべてはい**) をクリックするとデフォルトフォームが作成されます。デザインモードでこれらのフォームを修正したり、より洗練されたフォームで置き換えることができます。しかし、変更を一切行わなくても、これらのフォームを使い始めて、データベースにデータを登録したり表示したりすることができます。**いいえ**をクリックするとテーブルフォームは作成されません。この場合、データ入力や表示はプロジェクトフォームで行うことになります。テーブルごとに32,000までのフォームを使用し、それぞれ特定のタスクを行わせることができます。

#### プロジェクトフォームの特徴

プロジェクトフォームは以下の点でテーブルフォームと異なります:

- プロジェクトフォームは詳細 (ページ) タイプでのみ使用できます。リストフォームのメカニズムはプロジェクトフォームと互換性がありません。
- プロジェクトフォームはテーブルリストに表示されず、カレントの入力フォームや出力フォームとして指定できません。プロジェクトフォームをラベルや4Dの読み込み/書き出しエディター (データの読み込みと書き出し参照) で使用することはできません。
- プロジェクトフォームはDIALOGコマンドで表示するか、継承フォームとしてのみ使用できます (継承フォームを使用する参照)。
- プロジェクトフォームには、フィールドを含む、テーブルフォームと同じタイプのオブジェクトを配置できます。
フィールドがプロジェクトフォームに配置される場合、そのオブジェクトにはテーブル番号とフィールド番号が格納されます。フォームが他のデータベースやコンポーネントにコピーされると、参照もコピーされます。使用されるテーブルおよびフィールドはターゲットデータベースのものとなります。テーブ

ルが存在しなかったり、フィールドタイプが異なるなど非互換が発生した場合、フォームは正しく動作しません。
プロジェクトフォームは主にDIALOGコマンドのコンテキストで使用されることを想定しているため、(次レコードやレコード削除など) レコード管理の標準アクションボタンは、デフォルトではエディターやフォームウィザードに配置されません。レコードの表示やデータの変更はランゲージコマンドを使用して開発者が管理します。
他方、プロジェクトフォームをテーブルフォームの継承フォームとして使用する場合、自動レコード管理メカニズムを使用することが可能です。

- プロジェクトフォームには、テーブルフォームと同様、フォームメソッドを作成することができます。フォームメソッドにはエクスプローラーのCurrent form tableからアクセスできます。

#### テーブルフォームとプロジェクトフォームを相互に変換する

テーブルフォームをプロジェクトフォームに変更したり、プロジェクトフォームをテーブルフォームに変換したりすることが可能です。
テーブルフォームをプロジェクトフォームに変換すると、テーブルフォームで提供されていたデータ管理に関するすべての自動機能が動作しなくなるので、注意が必要です。同様に"リストフォーム"や"印刷用リストフォーム"タイプのフォームはページタイプのプロジェクトフォームに変換されます。

フォームタイプの変更はエクスプローラーのフォームページでドラッグ＆ドロップやコピー/ペーストを使用して行います。これは同じデータベース内あるいは2つのデータベース間で行えます。

プロジェクトフォームとテーブルフォームを相互に変換するには:

1. エクスプローラーのフォームページで、変換したいフォームを選択し、変換先項目上にドロップします。
   プロジェクトフォームをテーブルフォームに変換するときは、フォームを割り当てるテーブル名上にドロップします。
   同じデータベース内でフォームのドラッグ＆ドロップ操作を行うと、デフォルトでフォームが移動されます。フォームをコピーしたい場合はドラッグ＆ドロップの間**Alt** (Windows) あるいは **Option** (Mac OS) キーを押してください。2つのデータベース間でドラッグ＆ドロップを行うと、フォームはコピーされます。エクスプローラーのコンテキストメニューから**コピー**/**ペースト**コマンドを使用することもできます。

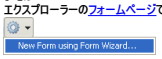
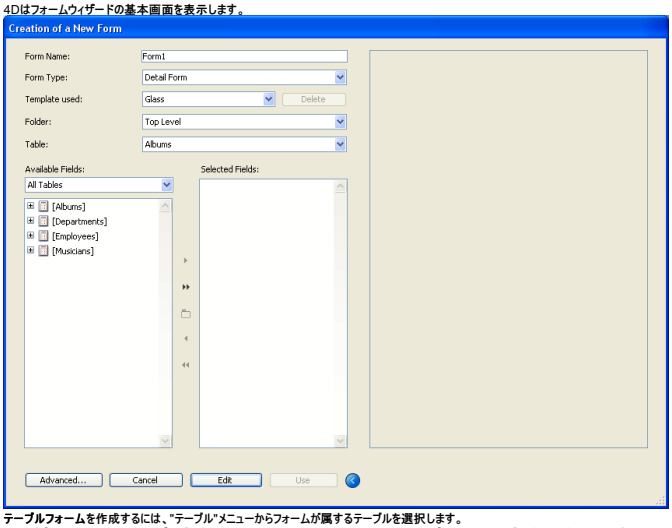
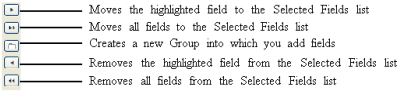
## フォームウィザードを使用してフォームを作成する

フォームウィザードを使用して新規フォームを手早く作成することができます。新規フォームを作成後すぐにそのフォームを使用したり、またはフォームエディターでフォームを編集することもできます。

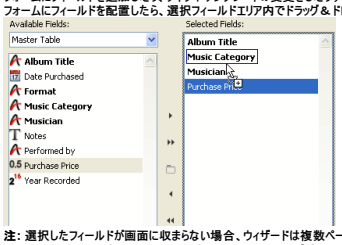
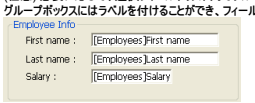
フォームウィザードには2 つの画面があります。基本画面を使用すると簡単な操作で新規フォームを作成できます。詳細画面ではフォームを作成する前にカスタマイズを加えることができます。

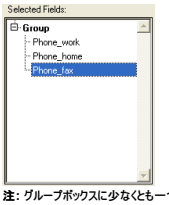
**基本的なフォームを作成する手順**

フォームウィザードで新しいフォームを作成するには:

1. **ファイル**メニューの**新規**またはツールバーの**新規**ボタンから**フォーム...**を選択します。
   または
   エクスプローラーのフォームページでオプションメニューをクリックし、**フォームウィザード...**を選択します。
   
   
   4Dはフォームウィザードの基本画面を表示します。
   
   
2. **テーブルフォーム**を作成するには、"テーブル"メニューからフォームが属するテーブルを選択します。
   ドロップダウンリストでマスターテーブルが選択されている場合、利用可能フィールド欄にテーブルのフィールドが文字順で表示されます。
   または
   **プロジェクトフォーム**を作成するには、テーブルメニューで**なし (プロジェクトフォーム)** を選択します。
   利用可能フィールドにはすべてのテーブルのフィールドがリストされます。リストフォームと印刷用リストフォームがフォームタイプメニューから取り除かれます。
3. フォーム名エリアにフォームの名前を入力します。
   ランゲージからは名前を使用してフォームを参照します。4Dの命名規則に準拠したフォーム名を使用してください (4Dランゲージリファレンスマニュアルの識別子参照)。
4. フォームタイプドロップダウンリストからフォームタイプを選択します。
   選択可能な項目:
   - **詳細フォーム**: データ入力や更新のためのフォーム
   - **リストフォーム** (テーブルフォームのみ): 画面にレコードを一覧表示するためのフォーム
   - **印刷用詳細フォーム**: 請求書のようにページごとに一レコードを印刷するためのフォーム
   - **印刷用リストフォーム** (テーブルフォームのみ): レコードリストを印刷するためのフォーム
5. フォームのテンプレートを選択します。
   テンプレートはフォント属性やフィールドラベルの位置、フィールドの周りを囲む四角の形状、プラットフォームインターフェースなどフォームアピアランスのいくつかの要素を制御します。4Dには組み込みのテンプレートがあり、また開発者はフォームウィザードを使用してカスタムテンプレートをテンプレートリストに追加することができます。カスタムテンプレートの追加についてはこの節の最後、"フォームテンプレートの作成"を参照してください。
6. (任意) フォームを配置するフォルダーを選択します。
   フォルダー名をドロップダウンリストから選択すると、フォームはこのフォルダーに含められます。アプリケーションではフォルダーを使用してオブジェクトをグループ化することができます。フォルダーはエクスプローラーのホームページで管理されます。デフォルトでフォームはトップレベルに作成されます。
7. フォームに配置するフィールドを"利用可能フィールド"エリアから選択します。
   フィールド名をダブルクリックするか、ドラッグ＆ドロップを使用する、またはパネルの中央にあるボタンを使用します:
   
   

   BLOBフィールドを除くすべてのフィールド属を選択できます。以下のフィールドを含むフォームを作成できます:
   - マスターテーブル (テーブルフォームの場合)
   - リレートした1テーブル
   - すべてのテーブル

   詳細設定オプションのサブフォームページ画面では、Nテーブルあるいはリレートされていないテーブルのフィールドを表示するサブフォームを配置することができます。
   マスターテーブルでないテーブルのフィールド、または自動1リレートテーブルでないフィールドを選択した場合、開発者がランゲージを使用して選択したフィールドの表示やデータ入力を制御しなければなりません。
   フォームにフィールドを追加したり、タイプやテンプレートの変更をしたりすると、ウィザードの右側にあるプレビューエリアにそれが反映されます。
   フォームにフィールドを配置したら、選択フィールドエリア内でドラッグ＆ドロップを使用して順番を変更できます:
   
   
   **注:** 選択したフィールドが画面に収まらない場合、ウィザードは複数ページフォームを作成し、すべてのボタンとスタティックオブジェクトをOページに配置します。このタイプのフォームを編集する場合、これらのオブジェクトを編集するにはOページを表示する必要があります。
8. (任意) 必要に応じて、選択フィールドリスト内でグループボックスを作成します。
   グループボックスにはラベルを付けることができ、フィールドをグループ化するために使用されます。グループボックスは以下のように表示されます:
   
   
   グループボックスを作成するにはグループボックスボタン をクリックします。選択フィールドリストに項目が追加されます。項目はデフォルトで"グループ"というラベルがつけられ、必要に応じて変更できます。
   **注:** グループボックスラベルはスタティックテキストです。他の4Dラベルと同様"ローカライズ"参照を使用することができます (スタティックテキスト中で参照を使用するとMissingRef参照)。
   利用可能フィールドからフィールドをドラッグし、グループボックス上にドロップすることで、グループボックスにフィールドを追加できます。
   選択フィールドリストは階層リスト形式で表示されます。グループボックスにフィールドが追加されると、それらはグループボックスの子要素として表示されます。




**注:** グループボックスに少なくとも一つの項目が存在し、階層が展開されていれば、選択フィールドにすでに追加されているフィールドをグループボックスの子要素上にドラッグ＆ドロップして追加することもできます。

9. 新規に作成するフォームをフォームエディターで編集するには**編集**ボタンをクリックします。
   または
   フォームをテストしたい場合は**使用**をクリックします (テーブルフォームのみ)。
   または
   フォームウィザードの詳細オプションを使用してさらにカスタマイズを行いたい場合は**詳細設定...**をクリックします。

## 詳細設定オプション

フォームウィザードの詳細設定オプションを表示するには、フォームウィザードの基本画面で**詳細設定...**ボタンをクリックします。フォームウィザードの最初の画面に戻るには**< 戻る**ボタンをクリックします。

詳細設定画面でもポイント＆クリック操作で新規フォームを作成できますが、より幅広いカスタマイズオプションが用意されています。

カスタマイズオプションはフォームウィザードの最初の画面で選択したフォータイプにより異なります。フォームウィザードでは以下のフォームタイプを指定できます:

- 詳細フォーム
- リストフォーム (テーブルフォームのみ)
- 印刷用詳細フォーム
- 印刷用リストフォーム (テーブルフォームのみ)

さらにフォームウィザードを使用して、カスタマイズオプションをテンプレートとして保存することができます。テンプレート名はフォームウィザードの基本ページにあるテンプレートドロップダウンリストに表示されます。ユーザー定義テンプレートを使用すれば、高度にカスタマイズされたフォームを、フォームウィザードの基本ページでカスタムテンプレートとフィールドを指定するだけで作成できます。

#### フィールドページ

フィールドページはフォームウィザードの基本画面と同じような外観をしています。フィールドページでは、フォームウィザードの基本ページと全く同じ方法で、フォームにフィールドを追加するために使用されます。この機能は基本画面をスキップして、詳細画面ですべての設定を行いたい開発者のために用意されています。フォームにフィールドを追加する方法については前節を参照してください。

#### リレートフィールド入力可

フィールドページには追加のオプション、**リレートフィールド入力可**があります。このオプションを使用して、リレートテーブルのフィールドの"入力可"プロパティ値を指定できます。このオプションはデフォルトでチェックされていて、フィールドは入力可です。例えばリレーションプロパティの"サブフォームにリレート値を自動代入する"が選択されている場合などに、ユーザーがリレートフィールドの値を更新できないようにするためにこのオプションの選択を外すことができます (リレーションプロパティについてはリレーションプロパティを参照)。
"入力可"プロパティについては入力可/必須入力とフィールドプロパティを参照してください。

#### オプションページ

オプションページではフォームサイズ、フィールドラベルの位置、フォームの表示に関する様々なオプションを設定できます。

#### フォームサイズ

このエリアではフォームサイズを指定できます。内容に合わせてフォームサイズを自動設定したり、最大幅と最大高さを入力して固定サイズにしたり、画面サイズを選択したりすることができます。これらの設定を組み合わせることも可能です。

画面サイズドロップダウンメニューでは以下の選択肢が提供されます:

- 自動 (フォームサイズは現在のスクリーンに合わせられます)
- 低解像度 640x480
- 中解像度 800x600
- 標準 1024x768
- 高解像度 1280x1024
- 大スクリーン 1600x1200

値はピクセル単位で、幅x高さの率に対応します。

画面サイズを入力するか、ドロップダウンリストから選択すると、プレビューエリアが選択した内容を反映して更新されます。フォームウィザードはフォーム上のフィールドやオブジェクトの配置を調整し、すべてのオブジェクトが選択されたスクリーンサイズに収まるようにします。**必要に応じてページを作成**オプション (後述) が選択されていて、4Dがすべてのフィールドを1ページ収めることができない場合、すべてのフィールドをフォーム上に配置するために必要なだけページが作成されます。複数ページを生成すると、フォームウィザードはボタン、フォームタイトル、その他すべてのページで共通して使用されるオブジェクトを0ページに配置します。

- **フィールドに合わせる**: このチェックボックスにチェックすると、フォームウィザードはフィールドの周りの背景項目をフィールドに合わせてサイズ調整し、空のスペースがなくなるようにします。
- **印刷用フォーム**の場合 (印刷用詳細フォームや印刷用リストフォーム):
  - "フォームサイズ"エリアには**用紙設定...**ボタンがあり、クリックすると印刷時の用紙サイズ等を設定できる印刷設定ダイアログボックスが表示されます。4Dはフォームのサイズとプレビューエリアをここで選択したページフォーマットに調整します。
  - "画面表示オプション"エリアでは、フォームタイトル、ページ番号、印刷日や印刷時刻など、レポートの印刷時に挿入したいオブジェクトを指定することができます。
- **リストフォーム**タイプの場合: "フォームサイズ"エリアには**対象の幅**オプションがあります。
  このオプションにチェックしない場合、リストフォームの幅はフォームフィールドに合わせて自動で計算されます。このオプションにチェックして値を入力すると、ウィザードはフィールド幅を減らして指定された幅にすべてのフィールドを収めるようとします。**必要に応じて切り捨て**オプションにチェックすると、ウィザードはフォーム幅が指定された値よりも小さくなるように必要なだけフィールドを取り除きます。この二番目のオプションにチェックしない場合、フォーム幅は**対象の幅**オプションで指定された値より大きくなる場合があります。

#### ラベルの位置

オプションページのラベルの位置エリアでは、フィールドに対してどの位置にラベルを配置するかを指定できます。ラベルを表示する場合、フィールドの左あるいは上を選択できます。

#### 画面表示オプション

オプションページの画面表示オプションエリアでは、フォームに要素を追加し、また追加のオプションを設定することができます:

- **フォームタイトル**: フォーム上部にテーブル名をタイトルとして追加します。プロジェクトフォームではこのオプションを使用できません。
- **1行に1フィールド**: フィールドをすべて縦に並べる場合にチェックします。このオプションが選択されていないと、フォームウィザードは可能な限りフィールドを横に並べようとします。
- **必要に応じてページを作成**: フィールドが1ページに収まらない場合、フォームウィザードが自動で複数ページを作成するように指示します。このオプションを使用すると、フォームウィザードは全ページで共通に使用されるオブジェクトを背景ページ (0ページ) に配置します。
- **ダイナミックフィールド名を使用**: このオプションが選択されると、フィールド名やテーブル名は動的参照を使用してフォームに挿入されます。これによりフィールド名やテーブル名が変更された場合、自動でその変更がフォームにも反映されます。テーブル名やフィールド名はストラクチャーエディターあるいはSET FIELD TITLESやSET TABLE TITLESコマンドを使用して変更されます。詳細はスタティックテキスト中で参照を使用するを参照してください。
- **関連メニューバー**: フォームにメニューバーを関連付けるには、このオプションにチェックを入れ、メニューバーを選択します。詳細はフォームにメニューバーを割り当てるを参照してください。
- **レコード番号/レコード数**: フォームにカレントレコード番号とそうレコード数を表示する4D変数 (デフォルトで*vRecNum*という変数名) が追加されます。このオプションはプロジェクトフォームでは使用できません。

#### ボタンページ

ボタンページではフォームで使用されるボタンをカスタマイズできます。

**注:** このページは"印刷用詳細フォーム"と"印刷用リストフォーム"タイプのフォームでは表示されません。

印刷用でないリストおよび詳細フォームでは、ユーザーはボタンを使用してレコードへの変更を保存したりキャンセルしたり、あるいは複数ページフォームではページ間を移動したりします。
さらに、テーブルフォームにはレコード間を移動する自動アクションボタン (先頭レコード、最終レコード、次レコード、前レコード)、サブフォームにレコードを追加/削除するボタン、またはカレントレコードを削除するボタンを配置することもできます。

ボタンページではボタンのデザイン、ボタンのアクション、フォーム上でのボタンの位置、各ボタンのラベルなどを指定できます。

**注:** フォームエディターでボタンを追加したり、削除したり、位置を変更したり、また他のオブジェクトを追加したりクリックされたときに実行されるアクションを指定することができます。

#### ボタンファミリーと位置

ボタンファミリーとボタンの位置エリアではボタンのスタイルと位置を指定できます。
ボタンファミリードロップダウンリストでファミリーを選択し、[<]や[>]ボタンをクリックしてボタンをプレビューできます。

#### 自動ボタンアクション

4Dは一連の組み込みボタンアクションを提供しています。ボタンに組み込みボタンアクションを割り当てれば、ユーザーがそのボタンをクリックした際に実行させる動作を指定するメソッドを書く必要がありません。

利用できる組み込みアクションの数はフォームのカテゴリー (テーブルまたはプロジェクト) により異なります。例えばレコード間を移動するアクションはプロジェクトフォームでは使用できません。

詳細フォームの場合、フォームウィザードで以下の組み込みボタンアクションを利用できます:

- **OK**: 新規レコードの保存、または既存レコードへの変更の保存を行う。
- **キャンセル**: 新規レコードの破棄、または既存レコードへの変更の破棄を行う。
- **次ページ**, **前ページ**, **先頭ページ**, **最終ページ**: 複数ページフォームにおいて指定されたページを表示する。

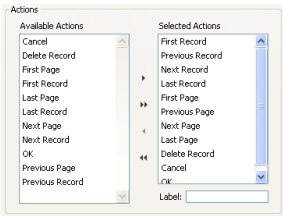
以下のアクションはテーブルフォームでのみ使用できます:

- **次レコード**、**前レコード**、**先頭レコード**、**最終レコード**: カレントレコードを保存し、指定されたレコードを表示する。
- **レコード削除**: データベースからカレントレコードを削除する (確認ダイアログボックスが表示される)。

**注:** サブフォームを挿入するとき、サブフォームページのオプションエリアで追加と削除ボタンんチェックソックスにチェックを入れておくと、4Dは自動で2つのサブフォームボタンを挿入します。**追加**はNテーブルまたはサブテーブルに新規レコードを追加し、**削除**はサブフォーム中で現在選択されているレコードを削除します。

4Dにはほかにもアクションが定義されたボタンがあります。これらのアクションはフォームウィザードでフォームを作成する際やフォームエディターでフォームを編集する際に設定できます。詳細は標準アクションを参照してください。

デフォルトボタンはページのアクションエリアに表示されています:




自動ボタンの選択や選択解除は、フィールドページでフォームにフィールドを配置したり取り除いたりするのと同じ方法で行います。追加したボタンは選択されたアクションエリアに表示されます。

**注:** "選択されたエリア"ボタンに含まれている場合でも、ページ管理ボタン (前ページや次ページ) は複数ページフォームが作成される場合にのみ実際に配置されます。

ボタンのデフォルトラベルを変更したい場合 (ラベルありテンプレートが選択されていなければなりません)、選択されたアクションエリアでボタンを選択し、ラベルエリアに新しいラベルを入力します。ラベル入力後、タブを押すか、選択されたアクションリストの他のボタンを選択します。入力したラベルがプレビューエリアに表示されます。

**注:**

- データベースローカライズのために、リソース参照をラベルとして入力することができます (付録 C: XLIFFアーキテクチャー参照)。
- ボタンTipはボタンラベルとは独立しています。ボタンにTipを割り当てたい場合、フォームエディターのプロパティリストを使用します (ヘルプメッセージ参照)。

#### サブフォームページ

サブフォームページではフォームにサブフォームを追加できます。このサブフォームは、サブフォームの自動更新メカニズムを使用したい場合、リレートしたNテーブルのフォームでなければなりません。

リレートしたNテーブルのフィールドを使用したい場合、フォームにサブフォームを追加します。サブフォームは複数のレコードを一度に表示します。サブフォームを使用してリレートしたレコードや他のテーブルのレコードを表示させることができます。サブフォームに表示されているレコードに情報を入力することもできます。

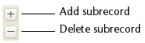
リレートしたNテーブル、またはリレートしていないテーブルのフィールドをサブフォーム中のフィールドに表示できます。リレートしたNテーブルのフィールドを配置した場合、リレーションによりどのレコードを表示するかが決定されます。リレートしていないフィールドやマニュアルリレーションのフィールドをサブフォームに配置した場合、デフォルトでテーブルのカレントセレクションが表示されます。メソッドを使用してレコードセレクションを制御できます。

詳細画面のサブフォームページでは、フォームをサブフォームとして使用し、サブフォームオプションを設定して、サブフォームに対して動作するボタンを追加することができます。

フォームにサブフォームを追加するには、**サブフォーム組み込み**オプションにチェックします。サブフォームのテーブルを"テーブル"ドロップダウンリストから選択し、リストフォームドロップダウンリストから使用するサブフォームを選択します。また"詳細フォーム"ドロップダウンリストから、ユーザーがリストサブフォームをダブルクリックした際に表示されるページフォームを設定することもできます。設定したサブフォームはカレントフォームの一部としてプレビューエリアに表示されます。

サブフォームには以下のオプションを設定できます:

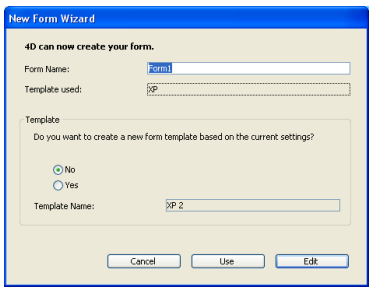
- **複数選択可**: ユーザーは**Shift** (連続選択)、あるいは**Ctrl** (Windows) や **Command** (Mac OS) (非連続選択) キーを使用して、同時に複数のレコードを選択することができます。
- **リスト入力可**: ユーザーはサブフォームに直接サブレコードの値を入力して、更新できます。
- **追加・削除ボタン**: ウィザードは自動で2つのボタンをフォームに挿入します (フォームで指定されたボタンファミリーが使用されます)。これらのボタンにはサブレコード追加とサブレコード削除の標準アクションが割り当てられます:




これらのオプションについてはリストサブフォームを参照してください。

### 詳細設定フォームを生成する

詳細設定画面で新しいフォームのプロパティをすべて設定したら、**OK**ボタンをクリックします。OKをクリックすると以下のダイアログボックスが表示されます:




新規フォームを作成するには**使用**ボタンをクリックしてフォームのテストを行うか (テーブルフォームのみ)、**編集**ボタンをクリックして新規フォームをフォームエディターで開きます。

#### フォームテンプレートを作成する

フォーム作成ダイアログでは、詳細設定画面で設定した内容に基づき、新しいフォームテンプレートを作成することができます。フォームテンプレートを作成すると、その名前がフォームウィザードのテンプレートドロップダウンリストに追加されます。フォームテンプレートはフォームとは別に保存されます。

フォームテンプレートを作成するには**はい**をクリックし、テンプレート名エリアに名前を入力します。

## 空のフォームを作成する

フォームウィザードを使用せずに、空のフォームを直接エクスプローラーで作成することができます。この場合、フォームはフィールドやボタン、変数が一切含まれない状態で作成され、フォームエディタ上で開かれます。この後フォームエディタ上ですべてを作成することができます。変数やプラグインエリアだけを含むダイアログボックスを作成したい場合には、この空白フォームを作成すると便利です。

空のフォームを作成するには:

1. エクスプローラーのフォームページを表示する。
2. 作成するフォームカテゴリーに対応する項目を選択する:
   - プロジェクトフォームの場合: "プロジェクトフォーム"または既存のプロジェクトフォームを選択。
   - テーブルフォームの場合: テーブル名またはそのテーブルに属するテーブルフォームを選択。
3. エクスプローラーの追加ボタンをクリック。以下のダイアログボックスが表示されます:
   - プロジェクトフォーム:
   - テーブルフォーム:
4. 入力エリアにフォーム名を入力。
   このフォーム名はプログラミングにおいて使用されます。
5. (任意) フォームを格納するフォルダーを選択。
   ドロップダウンリストからフォルダーを選択すると、フォームはこのフォルダーに配置されます。フォルダーを使用してエクスプローラーのホームページでアプリケーション内のオブジェクトをグループ化し、管理できます。デフォルトでフォームはトップレベルに作成されます。
6. **OK**をクリックします。
   空のフォームがフォームエディターで開かれます。

## フォームを編集する

以下のようにしてフォームエディターでフォームを開きます:

1. **ファイル**メニューから**開く > フォーム...** を選択する。
   または
   ストラクチャーエディターで、**Shift + Ctrl**を押しながら、開きたいフォームのテーブル名をダブルクリックする。
   または
   ストラクチャーエディター上でテーブル名を右クリックし、コンテキストメニューから**フォーム表示 ...**を選択する。
   4Dはエクスプローラーの[フォームページ]を表示します。プロジェクトフォームやテーブル階層リストを展開して、各項目に割り当てられたフォームを表示させることができます。ストラクチャーエディターでテーブル名をダブルクリックまたはコンテキストメニューを使用した場合は、そのテーブルが既に選択されています。
2. 編集するフォームの名前を右クリックして、コンテキストメニューから**フォーム編集 ...**を選択。
   または
   フォームを選択し、エクスプローラーのオプションメニューから**フォーム編集 ...**を選択。
   または
   フォーム名またはプレビューエリア上でダブルクリック。
   4Dはフォームエディター上でフォームを開きます。
   **注:** これらのアクションはエクスプローラーの[ホームページ]でも行うことができます。
   **4D Server:** 2人以上のユーザーが同時に同じフォームを編集しようとすると、オブジェクトのロックが発生します。あるユーザーがデザインモードでフォームを編集中、そのフォームはロックされます。他のユーザーは、最初のユーザーがフォームを閉じてロックを解除するまで、同じフォームを更新することはできません。

## フォーム名を変更する

プロパティリストあるいはエクスプローラーのフォームページでフォーム名を変更できます。

フォーム名はリストや詳細フォームを設定するとき、およびFORM SET INPUTやFORM SET OUTPUTなどフォーム名を受け入れるコマンドのパラメターとして使用されます。.

プロジェクトフォーム内やフォームが属するテーブル内で、同じ名前を使用することはできません。同じ名前が使用されていると、名前でフォームが決定できなくなってしまうからです。テーブルが異なる場合は同じ名前を使用できます。たとえばすべてのテーブルにおいて"Input"や"Output"という名前のフォームを作成できます。

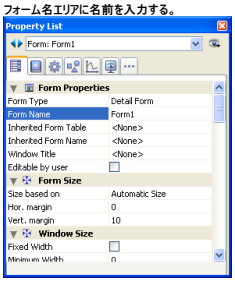
**警告**: (メソッドなど) データベース中のどこかで参照されるフォーム名を変更する場合、このフォームへの参照もすべて更新しなければなりません。これを行うために、デザインモードの検索および置換機能を使用することができます (名称変更参照)。

エクスプローラーでフォーム名を変更するには:

1. エクスプローラーのフォームページを表示する。
   テーブルとフォームの階層リストが表示されます。
2. **Alt** (Windows) または**Option** (Mac OS) を押しながらフォーム名をクリックする。
   または
   変更したいフォーム名を選択して、もう一度クリックする。
   フォーム名が編集状態になります。
3. 新しい名前を入力する。
4. 入力エリアの外側をクリックするかTabキーを押す。新しい名前が確定されます。

プロパティリスト (プロパティリストを使用する参照) を使用してフォーム名を変更するには:

1. プロパティリストにフォームのプロパティを表示させる。
2. フォーム名エリアに名前を入力する。




エクスプローラーから呼び出せるフォームプロパティウィンドウを使用してフォーム名を変更することもできます (フォームプロパティ (エクスプローラー)参照)。

## 詳細/一覧フォームを指定する

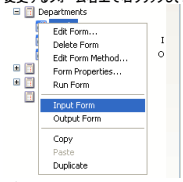
テーブルごとに1つづつカレント詳細 (入力) フォームとカレントリスト (出力) フォームがあります。詳細フォームは1レコードを表示し、レコードを作成、更新するために使用されます。リストフォームはレコードを一覧表示するために使用されます。

詳細フォームとリストフォームそれぞれについて、いつでもどのフォームを使用するか指定を変更できます。この変更はテーブルリストウィンドウまたはFORM SET INPUTやFORM SET OUTPUTなどのコマンドを使用して行います。この場合の変更はカレントセッションにのみ有効です。

デザインモードでデフォルトの詳細フォームやリストフォームを指定することもできます。この場合変更でデータベースに保存されます。

テーブルの詳細およびリストフォームを変更するには:

1. エクスプローラーのフォームページを表示する。
2. デフォルトの詳細およびリストフォームを変更するテーブルの階層を展開する。
   カレント詳細フォームの横には文字 I が、カレントリストフォームの横には文字 O が表示されます。
3. 変更するフォーム名上で右クリックし、コンテキストメニューから**入力フォーム**または**出力フォーム**コマンドを選択します:




   または
   エクスプローラーのオプションメニューから**入力フォーム**または**出力フォーム**コマンドを選択します。

**注:** 入力フォームや出力フォームの指定はテーブルフォームに対してのみ可能です。

1つのフォームに対し、入力フォームと出力フォーム両方を指定することもできます。この場合文字 B (Both) がフォーム名の横に表示されます。

## フォームを削除する

プロジェクトフォームや、カレントの詳細フォームでもリストフォームでもないフォームは、削除することができます。カレント詳細フォームやリストフォームを選択している場合、削除ボタンは使用できません。

フォームを削除するには:

1. エクスプローラーのフォームページを表示する。
2. プロジェクトフォームテーマあるいは削除するフォームが属しているテーブルの階層を展開する。
3. 削除するフォームを選択して削除ボタンをクリックする。
   または
   エクスプローラーのコンテキストメニューから**フォーム削除**コマンドを選択する (フォーム名上で右クリック)。
   4Dは削除を行うか確認してきます。
4. **OK**をクリックする。
   4Dはフォームを削除します。削除されたフォームはゴミ箱に移動され、ゴミ箱を空にしない限り復旧させることができます (ゴミ箱ページ 参照)。

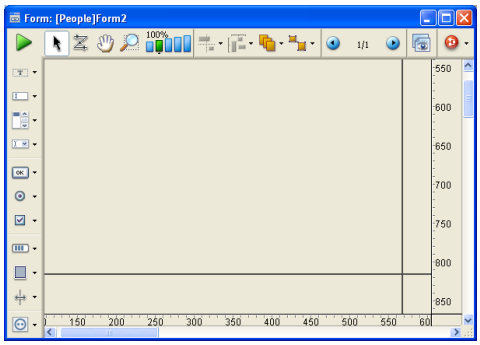
**注:** エクスプローラーのホームページで (同じ方法を使用して) フォームを削除することもできます。

## フォームを印刷する

それぞれのフォームの最大面積はおよそ125平方メートルです。スクロールを行って、フォームエディターに表示しきれない部分を表示することができます。画面表示用の場合、このエリア全体をフォームデザインで使用できます。フォームに配置した要素を表示するにはスクロールを行います。

印刷用途としては、フォームの要素は1ページの幅に収まっていなければなりません。しかしページの長さは複数ページ分にわたることができます。実際のページサイズは印刷機器、用紙サイズ、用紙設定ダイアログでの設定により異なります。4Dはフォームエディター上にページの境界線を表示します。これらの線はページの境界を示しています。ページの境界は用紙設定を変更すると書き換えられます。用紙設定の設定内容は、エディターが閉じられると保存されます。フォームの境界はフォームエディターのコンテキストメニューから**表示 > 用紙**と選択することで、表示と非表示を切り替えることができます (フォームエディター上での要素の表示/非表示参照)。

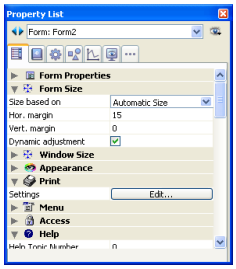
以下の図はページの境界線を示しています:




### フォームの用紙設定

フォーム毎に用紙設定を行うことができます。この設定はアプリケーションモードでフォームが印刷される際に使用されます。

用紙設定を行うには、フォームのプロパティリスト中、印刷テーマの**設定/編集...**ボタンをクリックします (フォームプロパティ参照)。




用紙設定ダイアログボックスが表示されます。このダイアログで用紙や向きなどフォームの用紙設定を変更します。この標準ダイアログで利用可能なオプションはシステムの設定により異なります。

アプリケーションモードでのレコード印刷に関する詳細はレコードを印刷するを参照してください。

## フォームを保存する

フォームに対して行った変更は保存しなければなりません。保存はフォームエディターを閉じたときと、保存メニューを選択した場合に行われます。フォームエディターはクローズボックスをクリックするか、**ファイル**メニューから**閉じる : フォーム : フォーム名**を選択します。エディターを閉じずに保存するには、ファイルメニューから**保存 : フォーム : フォーム名**を選択します。

フォームが保存されたら、引き続き作業を行うことができます。編集内容を元に戻したい場合、最後に保存された状態まで戻すことができます。これを行うには、ファイルメニューから**元に戻す ...**を選択します。

**4D Server:** デザインモードでフォームが保存されると、ユーザーが次回そのフォームを開く際に変更が有効になります。

## フォームの作成


フォームエディター

フォームプロパティ

フォームプロパティ (エクスプローラー)

複数ページのフォームを作成する

継承フォームを使用する

データの入力順を変更する

フォームオブジェクトの挿入と配置

オブジェクトの表示プロパティを設定する

テキストの回転

スタティックピクチャーを使用する

スタティックテキスト中で参照を使用する

グリッド上にオブジェクト作成

変数名を一括で連番にする

フォームのスケール

オブジェクトメソッドを使用する

オブジェクトビューを使用する

バッジを使用する

設定済みライブラリを使用する

カスタムオブジェクトライブラリの作成と使用

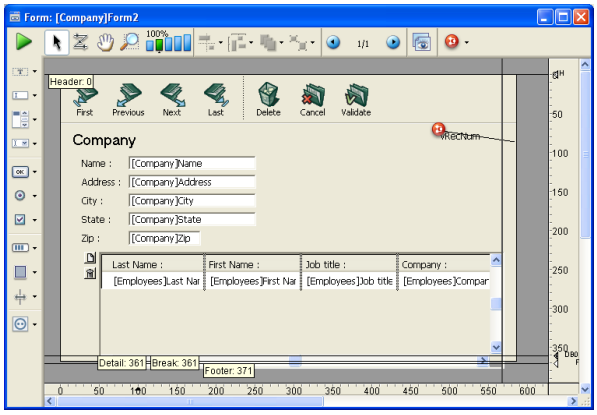
フォーム上のオブジェクトに関する情報を実行時に表示する

# フォームエディター

フォームウィザードでフォームを作成さいには多くのカスタマイズオプションがあります。テンプレートを使用してフィールドやフィールドラベルのアピアランスやフォームサイズの選択、配置する自動アクションボタンを制御できます。

しかしこれはカスタマイズの始まりにすぎません。4Dが提供するフォームエディターを使用すれば、必要とされる機能に達するまで完全にフォームをカスタマイズできます。フォームエディターではオブジェクトの作成や削除、オブジェクトの操作、フォームやオブジェクトのプロパティの設定が行えます。

フォームエディター利用時には、フォームごと個別にウィンドウが開かれ、ウィンドウごとにオブジェクトパートツールバーがあります。複数のフォームを同時に開くことができます。右と下にあるルーラーが、オブジェクトの配置を手助けします。ルーラの単位はセンチ、ピクセル、インチに変更できます。

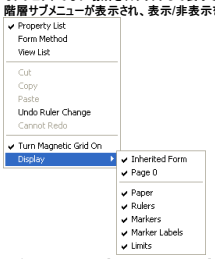



この節ではフォームエディターの基本要素をカバーします。アクティブオブジェクトを使用するではフィールドや他のアクティブオブジェクトの詳細について説明しています。

### フォームエディター上での要素の表示/非表示

フォームエディターの大部分のインタフェース要素は表示したり非表示にしたりすることができます。この機能により、フォーム上に作成または表示しなければならない要素や、使用したいツールだけを表示することができます。このオプションは常にフォームエディターのカレントウィンドウに適用されます。例えば出力フォームの作業を行う際に、出力コントロールラインを表示しておくと便利です。

フォームエディターで任意の要素を表示、また隠すには:

1. **フォーム**メニューから**表示**を選択します。
   または
   オブジェクトでない場所を右クリックして表示されるコンテキストメニューで、表示コマンドを使用する。
   階層サブメニューが表示され、表示/非表示を切り替えることのできる要素がリストされます:
   
   
   要素にチェックマークが付けられているものが表示されます。要素を隠すには、その要素を選択し、チェックマークを外します。
2. 表示、または非表示にしたい要素を選択する。
   このメニューのコマンドについて以下説明します:
   - **継承されたフォーム**: フォームのカレントページ上に、継承されたフォームのオブジェクト(継承されたフォームが存在する場合)を表示または非表示にします。詳細は"継承されたフォーム"を参照してください。
   - **ページ0**: フォームのカレントページにおいて、ページ0 のオブジェクトを表示または非表示にします。このオプションを選択すると、そのフォームのカレントページのオブジェクトとページ0 のオブジェクトとを区別することができます。フォームのページに関する詳細は"複数ページフォームを作成する"を参照してください。
   - **用紙**: 印刷ページの用紙境界を示す灰色の線を表示/非表示にします。後述の**境界**オプションが選択されている場合、このオプションは表示上明白な効果を持ちません。フォームのサイズが印刷ページより小さければ、用紙境界はフォームの外側に表示されます。フォームを印刷するを参照してください。
   - **ルーラー**: フォームエディターウィンドウのルーラーを表示/非表示にします。
   - **マーカー**: 出力フォームのエリアを識別する出力コントロールラインとマーカーを表示/非表示にします。
   - **マーカーラベル**: マーカーラベルを表示/非表示にします。これは出力コントロールラインが表示されている場合のみ有効です。詳細は"出力コントロールラインを移動する"を参照してください。
   - **境界**: フォームの境界を表示/非表示にします。このオプションが選択されていると、アプリケーションモードで表示されるとおりに、フォームがフォームエディターに表示されます。これによりアプリケーションモードに移動しなくてもフォームを調整しやすくなります。
     **注: サイズを決めるもの**, **水平マージン** そして **垂直マージン**フォームプロパティ設定はフォーム境界に影響します。これらの設定を使用すると、フォーム上のオブジェクトに基づいて境界を設定できます。フォームの境界を決定する位置にオブジェクトを配置したり、サイズを変更したりすると、境界も変更されます。フォームプロパティに関する詳細は"フォームプロパティの設定"を参照してください。

### 拡大/縮小表示

カレントフォームを拡大/縮小表示することができます。これを行うには虫眼鏡をクリックするか、直接拡大率をクリックします。拡大/縮小率は50%, 100%, 200%, 400% そして 800%です。

- 虫眼鏡ボタンをクリックすると、カーソルも虫眼鏡に変わります。フォーム中をクリックすれば拡大表示され、**Shift**を押しながらクリックすれば縮小表示されます。
- 拡大率バーをクリックすると、即座に表示が変更されます。

拡大/縮小表示中もフォームエディターの機能は利用できます。

### ツールバーを使用する

フォームエディターのツールバーはフォームを操作、更新するための一連のツールを提供します。ウィンドウごとに固有のツールバーをもちます。

ツールバーには以下の要素があります:

- **フォーム実行**: フォームの実行をテストするために使用します。このボタンをクリックすると、4Dは新しいウィンドウを開き、そのコンテキストでフォームを表示します (リストフォームの場合レコードリストを、詳細フォームの場合カレントレコードを)。フォームはメインプロセスで実行されます。
- **選択ツール**: フォームオブジェクトの選択、移動、リサイズを行います。詳細はオブジェクトを選択するを参照してください。
  **注**: テキストやグループボックスタイプのオブジェクトを選択すると、**Enter**キーを押すことで編集モードになります。
- **入力順**: フォームの現在の入力順を表示し、それを変更できます。詳細は"データ入力順"を参照してください。バッジを使用して入力順を確認することもできます ("バッジの使用"参照)。
- **移動**: 移動モードに移行し、ウィンドウ中をドラッグ＆ドロップすることで素早くフォームの表示部分を移動することができます。このモードでカーソルは手の形になります。このモードは特にフォームを拡大表示している時に便利です。
- **拡大/縮小**: フォーム表示の拡大/縮小率を変更できます (デフォルトで100%)。"拡大/縮小"モードにするには虫眼鏡をクリックするか、拡大/縮小率バーをクリックします。この機能は前節で説明しています。
- **整列**: このボタンにはフォーム中でオブジェクトの並びを揃えるためのメニューがリンクされています。このボタンは選択されているオブジェクトに応じて有効/無効になります。
- **均等配置**: このボタンには、フォーム上のオブジェクトを均等に配置するためのメニューが関連付けられています。このボタンは選択されているオブジェクトに応じて有効/無効になります。
  **注**: オブジェクトの整列と均等配置に関する詳細はオブジェクトを整列するを参照してください。
- **レベル**: このボタンには、フォーム上のオブジェクトの階層を変更するためのメニューが関連付けられています。このボタンは選択されているオブジェクトに応じて有効/無効になります。詳細についてはオブジェクトを重ねるを参照してください。

- **グループ化/グループ解除**: このボタンには、フォーム上の選択オブジェクトのグループ化やグループ解除を行うためのメニューが関連付けられています。このボタンは選択されているオブジェクトに応じて有効/無効になります。詳細はオブジェクトをグループ化するを参照してください。
- **表示とページ管理**: このエリアを使用して、フォームページ間を移動したり、ページを追加することができます。フォームページを移動するには矢印ボタンをクリックするか、または中央のエリアをクリックすると現われるメニューから表示したいページを選択します。最終フォームが表示されている状態で、右矢印ボタンをクリックすると、4Dはページを追加します。詳細は複数ページフォームを作成するを参照してください。
- **ビュー管理**: このボタンは、ビューパレットの表示や非表示を行います。この機能についてはオブジェクトビューを使用するで説明しています。
- **バッジ表示**: このボタンをクリックするたびに、すべてのタイプのフォームバッジが順に表示されます。また、このボタンには表示するバッジタイプを直接選択できるメニューが関連付けられています。この機能は"バッジを使用する"で説明されています。
- **定義済みオブジェクトライブラリー**: このボタンは定義済みオブジェクトライブラリーを表示します。定義済みオブジェクトライブラリーは定義済みのプロパティを持つオブジェクトを多数提供します。詳細は設定済みライブラリを使用するを参照してください。
- **ユーザロック**: ツールバーの右側にこのアイコンが表示されている場合、そのフォームに対し"ユーザによる更新可"プロパティが設定されていることを示します。この場合フォームをそのまま編集することはできず、最初にこのアイコンをクリックしてロックを解除しなくてはなりません。詳細についてはユーザによる更新可を参照してください。

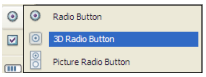
**オブジェクトバーを使用する**

オブジェクトバーには、4Dフォーム上で使用できるアクティブオブジェクトや非アクティブオブジェクトがすべて含まれています。一部のオブジェクトは、テーマ別にまとめられています(ボタン、ラジオボタン、メニュー等)。各テーマでは、複数の項目のなかから選択することができます。オブジェクトバーにフォーカスがある場合、キーボードのキーを使用してボタンを選択できます。以下の表で利用可能なオブジェクトグループとショートカットを示します。

| ボタン | グループ | キー |
|---|---|---|
| | テキスト / グループボックス | T |
| | フィールド / 変数 | F |
| | スクロールエリア / 階層リスト / リストボックス | L |
| | コンボボックス / ポップアップ/ドロップダウンリスト / 階層ポップアップメニュー / ピクチャーポップアップメニュー | P |
| | ボタン / 3Dボタン / ハイライトボタン / 非表示ボタン / ピクチャーボタン / ボタングリッド | B |
| | ラジオボタン / 3Dラジオボタン / ピクチャーラジオボタン | R |
| | チェックボックス / 3Dチェックボックス | C |
| | 進捗インジケーター / ダイアル / ルーラー | I |
| | 四角 / 線 / 過度の丸い四角 / 楕円 / 格子 | S |
| | スプリッター / タブコントロール | D |
| | プラグインエリア / サブフォーム / Webエリア | X |

任意のオブジェクトタイプを描画するには、該当するボタンを選択してから、フォーム上でそのオブジェクトを描きます。オブジェクトを作成した後でも、プロパティリストを用いてオブジェクトのタイプを変更することができます。強制的にオブジェクトを規則正しい形で描画するには、**Shift** キーを押しながらオブジェクトを作成します。この場合、線は水平方向、45 度、または垂直方向に引かれます。また、矩形は正方形に、楕円は正円に固定されます。

そのテーマで現在選択されているオブジェクトがフォームに挿入されます。ボタンの右側をクリックすると、バリエーションメニューが表示されます:




ボタンが選択されたら、**Shift+選択キー**のショートカットを使用して、各項目を順次移動することができます。ヘルプTip には現在選択されている項目と、そのオブジェクトに割り当てられた選択キーが表示されます。

ボタンを2 回クリックすると、フォーム上にオブジェクトを描画した後も、そのボタンが選択されたままになります(連続選択)。この機能により、同じタイプのオブジェクトを複数連続して作成しやすくなります。連続選択を解除したい場合は、別のオブジェクトやツールをクリックします。

**プロパティリストを使用する**

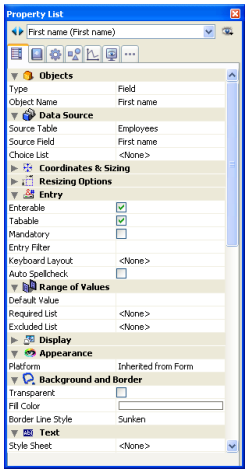
フォームとフォームオブジェクトにはプロパティが設定され、フォームへのアクセスやフォームの外観、およびフォーム使用時のフォームの動作が制御されます。例えば、フォームプロパティには、フォーム名、メニューバー、フォームサイズなどがあります。またオブジェクトプロパティには例えばオブジェクト名、オブジェクトサイズ、背景色、フォントなどがあります。

プロパティリストを使用してフォームおよびオブジェクトプロパティを表示・変更できます。エディターウィンドウでオブジェクト選択した場合はそのプロパティが、オブジェクトを選択しない場合はフォームのプロパティが表示されます。

プロパティリストを表示/非表示にするには:

1. **フォーム**メニューまたはフォームエディターのコンテキストメニュー (エディター中オブジェクト以外の場所で右クリック) から**プロパティリスト**を選択します。
   または
   **Ctrl+Shift+Space bar** (Windows) または **Command+Shift+Space bar** (Mac OS) ショートカットを使用する。

プロパティリストが表示されます:



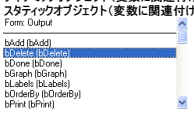

**Selection of contents**

プロパティリストには、フォームのプロパティまたは選択したオブジェクトのプロパティのいずれかが表示されます。:

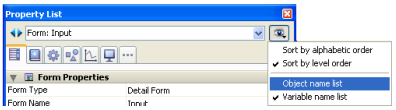
- オブジェクトが何も選択されていない場合、プロパティリストにはフォームのプロパティが表示されます。
- 1 つのオブジェクトまたは一連のオブジェクトが選択されている場合、プロパティリストにはそのオブジェクトのプロパティ、または選択された全オブジェクトに共通するプロパティが表示されます。

フォーム上の項目をクリックするか、プロパティリストウィンドウの一番上にある選択リストを使用して、プロパティを表示したいオブジェクトを選択することができます。この選択リストは3 つの部分に分かれており、線で区切られています。:

- フォーム名
- ダイナミックオブジェクト(変数に関連付けられる)
- スタティックオブジェクト(変数に関連付けられていない)




各ダイナミックオブジェクトに関して、このリストには変数名とその後にオブジェクト名が括弧で括られて表示されるか、またはオブジェクト名だけが表示されます。この表示内容は、リストに関連付けられたメニューを用いて指定することができます。メニューコマンドは現在表示されているものを反映しています:




また、このリストの各部分の並び順を変更することもできます。デフォルトでリストはレベル順（前面から背面へ）に並べられています。関連メニューを使用して、文字順にリストを並び替えることも可能です。

リストの左側にあるボタンを使用すると、リストの各オブジェクトを前面から背面へ向けて順に選択できます。

#### プロパティ表示ページ

ページ選択タブを使用し、プロパティリストに表示する情報を選択することができます。

- 最初のページ（メイン）には、選択オブジェクトに関連するすべての情報が各種テーマ（データソース、座標やサイズ、イベントなど）別に表示されます。テーマ名の左側にある展開アイコンをクリックすると、それぞれのテーマを展開したり、縮めることができます。これにより、操作したいプロパティだけを表示できるようになります。
  その表示はコンテキストに依存します。選択したオブジェクトまたはフォームのタイプに直接関連するテーマとプロパティだけが表示されます。また特定のプロパティの表示内容は、他のプロパティの値に応じて動的に変わります。例えばフィールドに対して入力可プロパティを設定すると、タブ有効プロパティが表示されます。
- 他の6 つのページにはそれぞれ、選択したオブジェクトに関連する特定の情報が含まれます。また、オブジェクトのタイプには関係なく、各テーマのプロパティがすべて表示され、関連のないプロパティはグレー表示されます。
  この補助的な方法により、任意のオブジェクトに指定された設定内容をより全体的に参照できるようになります。次に、6つのテーマページについて説明します:
  - 設定：オブジェクトの特性を定義するプロパティを含みます（タイプ、名前、データーソースなど）。
  - 動作：オブジェクトのダイナミック動作を定義するプロパティを含みます（関連メソッド、ドラッグ＆ドロップ、フォームイベント）。
  - サイズ：座標、サイズ、オブジェクトのリサイズを定義するプロパティを含みます。
  - 値：入力可オブジェクトの編集や表示を定義するプロパティを含みます（入力可、ショートカット、表示フォーマットなど）。
  - アピアランス：オブジェクトの外観を定義するプロパティを含みます。
  - 仕様：選択したオブジェクトタイプのプロパティを含みます（フィールドのヒント、アニメーションボタン、フォームの印刷設定など）。
  他のテーマページと異なり、そのオブジェクトと関係のないプロパティは表示されません。

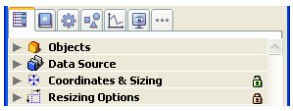
#### ナビゲーションショートカット

次のショートカットを使用し、プロパティリスト内を移動することができます:

- **矢印キー** ↑↓: あるセルから別のセルへ移動します。
- **矢印キー** ←→: テーマまたはメニューを展開／縮小します。
- **PgUpとPgDn**: プロパティリストの最初または最後の表示セルを選択します。
- **HomeとEnd**: プロパティリストの最初または最後のセルを選択します。
- イベント上で**Ctrl+click** (Windows) または **Command+click** (Mac OS) : クリックしたイベントの最初の状態に応じて、リストの各イベントを選択／選択解除します。
- テーマレベル上で**Ctrl+click** (Windows) または **Command+click** (Mac OS) : リストのすべてのテーマを展開／縮小します。

#### 南京錠アイコン

**ユーザによる更新可**プロパティがフォームに適用されている場合、南京錠アイコンがプロパティリストに表示され、ユーザによるプロパティの変更が可能かどうかを示します:




詳細はユーザによる編集可を参照してください。

### ルーラーを使用する

フォームにはルーラーがあります。ルーラーを非表示にして、フォームエディターの作業エリアを広くすることができます。必要であればいつでもルーラーを再表示できます。ルーラーの表示/非表示を切り替えるには、**フォーム**メニューまたはエディターのコンテキストメニューの**表示**サブメニューから**ルーラー**を選択します。

ルーラーには、オブジェクトを作成/サイズ変更時のポインターの位置が表示されます。オブジェクトを移動中はマーカーにオブジェクトの上下左右の位置が表示されます。オブジェクトマーカーを使用すると、ルーラー上で他のオブジェクトと同じ位置にそろえることができます。

好みに応じてルーラーの単位を変更できます。

ルーラー単位を指定するには:

1. フォームメニューから**ルーラー定義**を選択します。
   4Dは以下のようなルーラー定義ダイアログボックスを表示します。
2. 使用する単位を選択します。
   - **ポイント**を選択するとポイント単位でルーラーが表示されます。1ポイントは1ピクセルに幅に相当します。1インチは72ポイントです。
   - メートル法でルーラーを表示するには**センチメートル**を選択します。
   - ヤード法でルーラーを表示するには**インチ**を選択します。
3. **OK**をクリックします。
   4Dは選択した単位でルーラーを表示します。オブジェクトの座標も同じ単位を使用します。

# フォームプロパティ

フォームプロパティはフォームエディターのプロパティリストで設定します。
一部のフォームプロパティは、エクスプローラのプロパティウインドウで設定します。この件に関する詳細はフォームプロパティ (エクスプローラー)を参照してください。

## 一般的なプロパティ

#### フォームタイプ

フォームタイプを変更することができます。プロジェクトフォームの場合、**詳細フォーム**と**印刷用詳細フォーム**の2タイプが設定可能です。テーブルフォームの場合、さらに2つのタイプ、**リストフォーム**と**印刷用リストフォーム**が設定できます。

このプロパティはフォームのプロパティリストに表示されるオプションの種類を決定します。

またこの設定により、カレントの詳細/リストフォームリストに表示されるフォームの数を制限されます (テーブルリストウィンドウ、テーブルやフォームをブラウズする参照): リストに対応するタイプのフォームだけが表示されます。

フォームタイププロパティはプロパティリストの最上部にあります。フォームタイプに**なし**が設定されていると、そのフォームはテーブルリストの両方のメニューに表示されます。

**注:** エクスプローラーのフォームプロパティウインドウでフォームタイプを指定することもできます (フォームプロパティ (エクスプローラー)参照)。

#### 継承フォーム

このプロパティについては継承フォームを使用するで説明します。

#### デフォルトウィンドウタイトル

デフォルトのウィンドウタイトルはアプリケーションモードでOpen windowやOpen form windowを用いてフォームがを開かれた場合に使用されます。このデフォルトウィンドウタイトルはウィンドウのタイトルバーに表示されます。デフォルトのウィンドウタイトルを設定するには、プロパティリストのウィンドウタイトル入力エリアにタイトルを入力します。

動的参照を使用して、ウィンドウタイトルを定義することもできます:

- 標準のXLIFF参照 (付録 C: XLIFFアーキテクチャー参照)
- テーブル/フィールドラベル: 適用できるシンタックスは**<?[TableNum]FieldNum>** または **<?[TableName]FieldName>**です。参照はFORM SET INPUTコマンドの後にOpen windowまたはOpen form windowコマンドでウィンドウを開くときに解決されます。Open window場合はFORM SET INPUTコマンドを * 引数付きで実行しなければなりません。
- 変数またはフィールド: 適用できるシンタックスは **<VariableName>** または **<[TableName]FieldName>**です。フィールドや変数の現在の値がウィンドウタイトルとして表示されます。

**注:**

- ウィンドウタイトルの最大文字数は31です。
- エクスプローラーのフォームプロパティウインドウを使用してウィンドウタイトルを設定することもできます (フォームプロパティ (エクスプローラー)参照)。

#### ユーザーによる更新可

**ユーザーによる更新可**プロパティで、アプリケーションモードのユーザーフォームエディターを使用してユーザーがフォームをカスタマイズできるかどうかを指定します (ユーザーフォーム参照)。このメカニズムを使用する場合は追加の設定が必要です。これは4Dランゲージリファレンスのユーザフォームで説明しています。

**ユーザーによる更新可**プロパティにチェックされていると、フォームはロックされます。このプロパティが設定されたフォームをエディターで開くとツールバーに南京錠アイコンが表示され、そのフォームを更新することはできません。フォームのロックを解除するにはこのアイコンをクリックします。ただしこれを行うとユーザーフォームは無効となり、再生成しなければなりません。

**ユーザーによる更新可**プロパティにチェックされていると、プロパティリストのいくつかのオプションにも南京錠アイコンが表示され、ロックされたプロパティーが示されます。これらのプロパティはフォームをカスタマイズするユーザーに対してロックされています。

このアイコンをクリックして鍵を開いた状態にすると、ユーザーフォームエディターで対応するプロパティを編集できるようになります。

**注:** エクスプローラーのフォームプロパティウインドウでユーザーによる更新可プロパティを設定することもできます (フォームプロパティ (エクスプローラー)参照)。

#### 配置を保存

**配置を記憶**のオプションがチェックされていると、Open form window コマンドに * 演算子を渡して開かれたウィンドウに関して、セッション中にどのように変更されていてもそのウィンドウが閉じられたときにフォームの複数の情報が4Dによって自動的に保存されます:

- カレントページ
- それぞれのフォームの配置、大きさ、表示状態(リストボックス列のサイズと表示状態も含みます)。

**注:** このオプションは、OBJECT DUPLICATE コマンドを使用して作成されたオブジェクトに対しては無効です。このコマンドを使用しても使用環境を復元させるためには、デベロッパがオブジェクトの作成・定義・配置の手順をその都度行わなければなりません。

このオプションが選択されているとき、一部のオブジェクトに置いては**値を記憶**のオプションが選択可能になります(詳細は ウィンドウの配置の記憶 を参照して下さい)。

#### オブジェクトを反転させない (Windows)

**オブジェクトを反転させない**はフォームの"Right-to-left"モードを無効にします。"Right-to-left"モード (Windowsのみ) はデータベースのフォームオブジェクトやタイトルを反転させます。これはアラビア語など右から左に向けて読む言語のコンテキストで使用されます。このモードを有効化する詳細についてはOBJECT Get enabledの"Right-to-left言語"を参照してください。

**オブジェクトを反転させない**オプションがチェックされていると、データベース設定にかかわらず、アプリケーションモードでright-to-leftが適用されることはありません。このオプションは例えばright-to-leftモードに設定されたデータベースで"国際化された"フォームを設定するために使用されます。このオプションはプロパティリストおよび、フォームプロパティ (エクスプローラー)ダイアログボックスにあります。

**注:** このプロパティはFORM GET PARAMETERコマンドを使用してプログラムで管理することもできます。

## フォームとウィンドウのサイズ

フォームは常にウィンドウ内に表示されます。4Dではフォームおよびウィンドウのサイズ、およびそれらがサイズ変更された際の振る舞いをそれぞれ設定できます。
これらのプロパティは相互に作用し、アプリケーションのインターフェースはこれらの相互作用の結果となります。各フォームオブジェクトに設定されたリサイズオプションもインターフェースに影響を及ぼす点に留意してください (この点については"リサイズプロパティの設定"を参照してください)。

#### フォームサイズ

プロパティリストの"フォームサイズ"テーマでフォームサイズプロパティを設定します。以下の選択肢があります:

- **サイズを決めるもの** : **自動サイズ** : フォームサイズは、すべてのオブジェクトを表示するために必要なサイズと、**水平マージン**および**垂直マージン**フィールドへ入力されたマージン値(ピクセル単位)を合計したものになります。
- **サイズを決めるもの** : **サイズ設定** : フォームサイズは**幅**および**高さ**フィールドへ入力された値(ピクセル単位)により決まります(サイズ設定オプションを選択すると、水平マージンと垂直マージンフィールドが幅と高さフィールドに変わります)。
- **サイズを決めるもの** : **オブジェクト** : フォームサイズは、選択したフォームオブジェクトの位置により決まります。例えば、表示されるエリアの右下部分に置かれているオブジェクトを選択した場合は、左上端がフォームの起点であり、右下端が選択したオブジェクトの右下端となる矩形にマージン値を加算したものがフォームサイズになります。
  自動サイズのウインドウを用いて、オフスクリーンエリア(ウインドウの矩形境界線の外側のエリア)に配置したアクティブオブジェクトを使用したい場合ににこのオプションを選択することができます。このオプションを選択すると、これらのオブジェクトによりウインドウサイズが変更されなくなります。
- **自動サイズ**オプションまたはオブジェクトに基づいたサイズを選択すると、**水平マージン**と**垂直マージン**フィールドが表示されます。このフィールドに値を入力し(ピクセル単位)、フォームの端に追加するマージンを設定することができます。また、これらの値によりラベルで使用されるフォームの上マージンと右マージンも決まります。
  **注:** リストフォームの場合は水平マージンまたは幅フィールドだけが利用可能です。
- **常にフォームサイズを自動計算** : **自動サイズ**プロパティが選択されている場合、4Dバージョン2004より前のバージョンから変換されたフォームに対し、"フォームサイズ"テーマの"常にフォームサイズを自動計算"プロパティを利用することができます。このプロパティを使用して、フォームのリサイズモードを変更することができます。以前のバージョンの4Dでは、フォームに**自動サイズ**プロパティが設定されていると、フォームを開く時にのみフォームサイズが計算されていました。この後でOBJECT MOVEのようなコマンドを使用して変更を行っても、フォームサイズは調整されませんでした。この設定を使用すれば、このような場合でもフォームサイズが動的に調整されます。この処理方法は、新規フォームに対して有効です。一方で、互換性上の理由から、変換後のデータベースのフォームには、デフォルトとしてこのプロパティが選択されていません。変換後のフォームのサイズを動的に調整したい場合は、この**常にフォームサイズを自動計算**オプションを選択します。

#### ウィンドウサイズ

入力フォームをカスタムアプリケーションで表示する際、通常はOpen windowやOpen form window関数を使用してウィンドウを開きます。

Open windowを使用すると、ウィンドウのタイプとともに、ウィンドウの左、上、右、下の座標を指定することができます。この場合、ウィンドウのサイズはフォームのサイズに影響されません。一方で、フォームサイズテーマで設定したオプションとウィンドウのタイプによって、ウィンドウサイズを変更できるかどうかが決まります。Open form windowは引数に渡されたフォームのサイズとリサイズのプロパティに基づいて阿多rしいウィンドウを作成します。

プロパティリストでフォームウィンドウのサイズ変更について設定することができます。以下のオプションを使用できます:

- **制約付き**: このプロパティはバージョン2004より前の4Dから変換されたフォームで使用できます。これは以前の"サイズ変更可"フォームプロパティの振る舞いを再現するために使用されます。このオプションがチェックされていないと、オブジェクトのリサイズとウィンドウサイズの制約を処理するメカニズムは無効になります。この方法で、ユーザーは自由にフォームウィンドウをリサイズできますが、フォーム上のオブジェクトはリサイズも移動もさ

れません。最小/最大あるいは固定サイズプロパティ、およびオブジェクトのリサイズプロパティは無視されます。これは以前のバージョンの4Dの振る舞いに対応し、特別なインターフェースにおいて互換性維持の目的でのみ使用されるべきです。現在の4Dの振る舞いに対応させる場合はオプションにチェックします (標準モード)。

- **ウィンドウ幅固定**: このオプションを選択するとウィンドウ幅が固定され、ユーザーは変更できなくなります。このオプションを選択しない場合、フォームのウィンドウ幅を変更することができます。この場合、**最小幅**と**最大幅**入力エリアを使用し、変更できるサイズを制限できます。
- **ウィンドウ高さ固定**: このオプションを選択するとウィンドウの高さが固定され、ユーザーは変更できなくなります。このオプションを選択しない場合、フォームのウィンドウの高さを変更することができます。この場合**最小高さ**と**最大高さ**入力エリアを使用し、変更できるサイズを制限できます。

原則として、ユーザが入力可エリアや制御ボタンを隠せないようにする必要があります。

### アピアランス

#### プラットフォーム

プラットフォームインタフェースプロパティを使用し、実行コンテキストに合わせてフォームの外観を設定します。このプロパティは、フォームオブジェクトごと個別に設定することも可能です (LOG EVENTの"プラットフォーム"参照)。

**システム**および**印刷**という2 つのインタフェースプロパティを使用することができます。これら2つのプロパティは次のように使用します: 画面表示の際、フォームは現行のOS のインタフェースに準拠しなくてはなりません(システム)。一方、フォームが印刷用に設定されている場合は、プラットフォームとは関係なく、オブジェクトの外観を合わせておかなくてはなりません(印刷)。

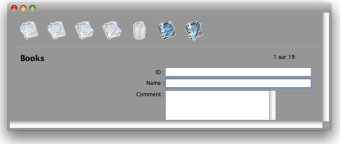
- **システム**: このプロパティを使用すると、4Dが稼働している現行のプラットフォームに応じて、フォームやオブジェクトの外観を自動的に合わせることができます:
  - Mac OS X上でフォームが表示される場合、フォームやオブジェクトはMac OS Xのルック&フィールになります。
  - Windows上でフォームが表示される場合、フォームやオブジェクトはコントロールパネルで設定されたアピアランスを使用します。

  4Dバージョン2004意向で作成されたデータベースのすべてのフォームとオブジェクトでこのモードがデフォルトで使用されます。
- **印刷**: このプロパティを使用すると、フォームやオブジェクトの外観を印刷用に設定できます。すべてのオブジェクトやグラフィック項目 (ボタン、チェックボックス、タブ、線等) はベクターとして描画され、印刷用に最適化されます。フォームが表示や印刷されるとき、プラットフォームにかかわらず同じ結果が得られます。

#### Mac OSアピアランスオプション

Mac OS用に2つの特別なオプション、**テクスチャー**と**ツールバーボタン**を使用してフォームアピアランスを設定できます。フォームがWindowsで実行された場合、これらのオプションの効果はありません。

- **テクスチャー (Mac OS)**: このフォームプロパティは、Mac OS 上でOpen form windowコマンドを使用してフォームが表示された時、そのフォームのテクスチャープロパティを有効にします。この外観はMac OS Xインターフェースを通して見ることができます:

  フォームがOpen form windowで作成されていないウィンドウ (例えばデザインモード) で表示された場合、このプロパティは考慮されません。Mac OSにおいて**テクスチャー**オプションが設定され、フォームの境界が表示されている時、フォームエディターにはメタルルックがプレビューされます。
- **ツールバーボタン (Mac OS)**: このプロパティを選択すると、Mac OSにおいて、ツールバー管理ボタンが表示されます。この標準ボタンはウィンドウのツールバーを表示したり非表示にしたりするために使用されます:

  このオプションはたとえばOpen form windowとDIALOGコマンドでダイアログを開く場合に使用します。これはOpen windowやOpen form windowコマンドでHas toolbar button Mac OS定数を使用するのと同じです。
  **注:** ユーザーがボタンをクリックすると、On Mac toolbar buttonフォームイベントが生成されます。
  ボタンがクリックされたときに実行されるアクションは開発者が実装しなければなりません。

#### グローボックスを隠す

このオプションにチェックを入れると、フォームウィンドウのグローボックスが非表示になります。このオプションはたとえばDIALOGコマンドを使用してフォームを呼び出した際に考慮されます。

### 印刷

印刷プロパティについてはフォームを印刷するで説明しています。

### フォームにメニューバーを割り当てる

アプリケーションを作成する際は、カスタムメニューを作成します。カスタムメニューを使用すると、メニューコマンドを追加して、レポート作成などのデータベース特定タスクを自動化することができます。

カスタムメニューはメニューバーエディターで作成します。作成するメニューバーには、それぞれ少なくとも1 つのメニューがあり、ユニークなID 番号と名前が割り当てられます。メニューバーやメニュー、およびメニュー項目の作成に関する詳細はメニューとメニューバーを参照してください。

フォームにメニューバーを割り当てるには、プロパティリストの"連結メニューバー"リストからメニューバーを選択します。**[...]**ボタンをクリックすると直接メニューバーエディターにアクセスできます (メニューとメニューバー参照)。アプリケーションモードで、フォームに割り当てられたメニューバーはカレントメニューバーの右側に追加されます。

フォームのメニューバーとカレントメニューバーが同じ場合、追加はされません。フォームメニューバーは詳細およびリストフォーム療法で動作します。

- メニューバーを選択すると**アクティブメニューバー**オプションが表示されます。このフォームをカスタムアプリケーションで使用したい場合、**アクティブメニューバー**オプションを選択します。このオプションは4Dにカレントメニューバーを無効にしないように指示します。このオプションが選択されていないと、4Dはカレントメニューバーを無効にし、フォームメニューバーへのアクセスのみが可能となります。

**注:** フォームプロパティ (エクスプローラー)ダイアログボックスでフォームにメニューバーを割り当てるkともできます。

### フォームアクセス権の制御

アクセス権とオーナーにユーザーグループを設定し、フォームへのアクセスを制御することができます。アクセス権およびオーナードロップダウンリストを用いて、それぞれの権限を1 つのグループに割り当てることができます。ユーザとグループによるパスワードアクセスシステムの作成に関する詳細はユーザー&グループを参照してください。

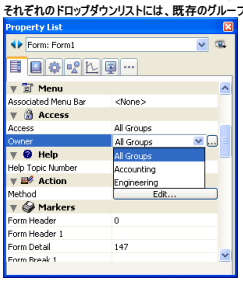
アクセス権ドロップダウンリストにより、レコード表示ウィンドウやカスタムアプリケーションでそのフォームを使用できるグループを制御します。そのグループに属さないユーザがこのフォームを利用しようとすると、そのユーザのパスワードではそのフォームを使用できない、というメッセージが4Dから表示されます。

オーナードロップダウンリストにより、デザインモードでそのフォームを編集できるグループを制御します。そのグループに属さないユーザがデザインモードでこのフォームを編集しようとすると、4Dは、そのユーザにはフォームを編集するためのアクセス権がない旨のメッセージを表示します。

両方のグループに指定されているユーザはデザインモードとカスタムアプリケーションでそのフォームを使用することができます。

フォームのアクセス権を設定するには:

1. プロパティリストでフォームプロパティを表示する。
2. **アクセス権**および**オーナー**ドロップダウンリストを使用して、希望するアクセス権を指定する。
   それぞれのドロップダウンリストには、既存のグループ名が表示されます。

   **[...]**ボタンを使用すると、4Dのツールボックス上で直接グループエディタにアクセスすることができます。

フォームプロパティ (エクスプローラー)ダイアログボックスを使用してアクセス権を割り当てることもできます。

### ヘルプ

4Dではデータベースごとにカスタムオンラインヘルプファイルを割り当てることができます。ヘルプファイルの作成については付録 A: カスタムヘルプファイルを割り当てるで説明しています。

ヘルプファイルは、それが呼び出されたコンテキストに応じて、関連する内容を表示させることができます。これを行うには**ヘルプトピック番号**をフォームに割り当てます。このフォームに割り当てるヘルプトピック番号はヘルプファイルに指定された番号と同じでなければなりません。これはプロパティリストの"ヘ

ルプテーマで行います。

**アクション**

**編集...** ボタンをクリックするとフォームメソッドがメソッドエディターで開かれます。この点については[メソッドの編集](#)で説明しています。

**マーカー**

このエリアを使用して、フォームの縦ルーラー上でのマーカーの位置を精密に指定することができます。マーカーは主に一覧フォームで使用されます。マーカーはリストされる情報を制御し、ヘッダーやブレーク、詳細、フッターを制御します。マーカーについては[出力コントロールラインを使用する](#)を参照してください。

**注：** 横ルーラー上のラベル幅を示す三角は、PRINT LABELコマンドを使用してラベルを印刷するためにフォームを作成する際のラベルの幅を制御します。

**イベント**

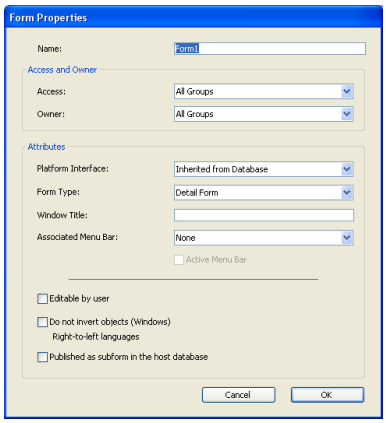
このエリアではフォームメソッドの実行を導くことのできるイベントを設定します。フォームが使用される際、ここで選択されたイベントだけが生成されます。
イベントを一切選択しない場合、フォームメソッドが呼び出されることはありません。
必要のないイベントの選択を解除することで、データベースの実行が早くなります。
フォームイベントについてはForm eventコマンドを参照してください。

すべてのイベントを一括して選択/選択解除するには、**Ctrl** (Windows) または **Command** (Mac OS) キーを押しながらイベントをクリックします。

## フォームプロパティ (エクスプローラー)

エクスプローラーのフォームページで主なフォームプロパティにアクセスできます。これを行うにはフォームを選択し、コンテキストメニューまたはページのオプションメニューから**フォームプロパティ...**コマンドを選択します。

プロパティがダイアログボックス内に表示されます:




このダイアログボックスで設定するプロパティのほとんど (名称、アクセス権、プラットフォームなど) は、フォームエディターのプロパティリストでも設定できます。これらはフォームプロパティで説明されています。他方、"ホストデータベース中でサブフォームとして公開"プロパティはこのダイアログボックスにのみあります。

**ホストデータベース中でサブフォームとして公開**

このオプションはフォームを含むコンポーネントの開発する際に使用します (フォームの共有参照)。

ホストデータベース中でコンポーネントフォームをサブフォームとして選択できるようにするためには、フォームのプロパティダイアログボックスで**ホストデータベース中でサブフォームとして公開**にチェックを入れ、明示的に公開フォームとして指定しなければなりません。プロジェクトフォームのみを公開フォームとして指定できます。

コンポーネントサブフォームの指定と利用に関する詳細は、コンポーネントの開発を参照してください。

## 複数ページのフォームを作成する

詳細フォームでは複数のページを作成することができます。一画面に納まりきらない数のフィールドや変数がある場合は、これらのオブジェクトを表示するためにフォームページを追加する必要があります。フォームに複数のページがあると、次の操作を行えるようになります:

- もっとも重要な情報を最初のページに配置し、他の情報を後ろのページに配置する。
- トピックごとに各ページにまとめる。
- データ入力中のスクロール動作を少なくしたり、または不要にする。
- フォーム要素の周りの空間を広げ、洗練された画面設計を行う。

複数のページは詳細フォームで使用する場合にのみ役立ちます。印刷やリストフォームには向きません。マルチページフォームを印刷すると、最初のページしか印刷されません。

フォームのページ数には制限がありません。フォーム内の複数ページ上に同じフィールドを何度でも表示することができます。しかしフォームページが多くなるほど、フォームの表示に要する時間が長くなります。

複数ページフォームには、1 つの背景ページと複数の表示ページが存在します。フォームエディターで背景ページ上に置かれたオブジェクトはすべての表示ページに現れますが、それらのオブジェクトの選択や編集は背景ページからしか行えません。複数ページフォームでは、ボタンパレットを背景ページに置くべきです。

またページ移動ツールオブジェクトを背景ページに配置し、ユーザに提供する必要があります。ページ移動ツールの追加に関する詳細はページナビゲーションコントロールを追加するを参照してください。

**注:** フォームウィザード詳細画面のオプションページには、選択したフィールドが1ページに収まらない場合、自動で複数ページを作成するオプションがあります。このオプションを選択すると複数の表示ページが、作成されるフォームに含まれるかもしれません。フォームウィザードが複数ページを作成する際、ボタンやフォームタイトル、外枠の四角などは背景ページ (0ページ) に置かれます。

### フォームに表示ページを追加する

それぞれのフォームには、少なくとも1 つの表示ページと背景ページがあります。カレントページ番号はフォームウィンドウのツールバーに表示されます。このエリアにはポップアップメニューも組み込まれ、このメニューを用いて表示ページを選択することができます。

表示ページを追加するには:

1. **フォーム**メニューから**ページ追加**を選択する。4Dは新しいページを追加します。
   または
   フォームの最終ページに移動し、ツールバーの次ページアイコンをクリックします。4Dはページを追加するかどうかを尋ねるダイアログボックスを表示します。**OK**をクリックして新しいページを作成します。
   新しい空の表示ページがフォームエディターウィンドウに表示されます。ウィンドウのページ番号エリアには現在のページ番号が表示されます。
   You can now add fields and other form elements to the new page.

これでフィールドやその他の要素を新しいページに配置できるようになります。

現在のページの前に新しいページを挿入するには:

1. **フォーム**メニューから**ページ挿入**を選択する。4Dは現在のページの前に新しいページを挿入し、スクリーンに表示します。

### ページ間を移動する

背景ページを表示したり別の表示ページへ移動したりしたい場合は、ツールバーのページ移動ツールを使用するか、またはフォームエディターウィンドウのページポップアップメニューを使用できます。

背景ページ(ページ0)を表示するには:

1. フォームの最初のページへ移動して、ツールバーの前ページアイコンをクリックする。
   または
   ページポップアップメニューを使用してページ0 へ移動する。
   または
   **フォーム**メニューの**ページ移動**サブメニューから0を選択する。
   または
   **Alt** キー(Windows)または**Option** キー(Mac OS)を押しながらページ0に属すオブジェクトをクリックするか、またはカレントページに属するオブジェクトの外側をクリックする。(カレントページに属するオブジェクト上でAlt+click や Option+clickを行うと、オブジェクトメソッドが作成され、開かれます。)
   4Dは背景ページを表示します。背景ページのページ番号は0です。背景ページに配置されたオブジェクトは各ページに表示されます。背景ページにはどのようなオブジェクトでも置くことができます。

**注: 表示**サブメニューの**ページ0**メニュー項目を使用しても、背景ページはカレント編集ページに設定されないため、この2 つを混同しないようにしてください。**表示**サブメニューの**ページ0**メニュー項目を選択しても、背景ページにある項目の表示/非表示がフォームエディター上で切り替わるだけです。詳細についてはフォームエディターのフォームエディタ要素の表示と非表示を参照してください。

ページナビゲーションツールを使用するには:

1. 次ページに移動するにはツールバーの次ページアイコン[#picture id="109343"/ language="fr"]をクリックします。
   または
   前ページに移動するにはツールバーの前ページアイコン[#picture id="109347"/ language="fr"]をクリックします。
   4Dは即座にカレントページの前後のページを表示します。

1ページを表示中に前ページに移動すると、背景ページに移動します。背景ページを表示中に前ページボタンをクリックしても何も起こりません。最後のページを表示中に次ページボタンをクリックすると、4Dはフォームに新しいページを作成するか尋ねてきます。

任意のページを表示するには:

1. ページ番号エリアをクリックし、マウスボタンを押したままにする。ポップアップメニューが表示されます。
   または
   **フォーム**メニューの**ページ移動**サブメニューを表示する
2. 移動先のページ番号を選択する。

### ページを削除する

複数ページフォームから不要な表示ページを削除することができます。削除されるページ上のフィールドやオブジェクトも一緒に削除されます。残りのページのページ番号は付け替えられます。1 番目のページと背景ページ(0 ページ)だけで構成されるフォームの場合は、いずれのページを削除することもできません。

フォームからページを削除するには:

1. ページ移動ツールまたはページポップアップメニューを使用して、削除したいページを表示する。
2. **フォーム**メニューから**ページ削除**を選択する。
   ダイアログボックスが表示され、そのページをフォームから削除するかどうかを確認してきます。
3. **OK**をクリックする。
   ページおよびそのページ上の全オブジェクトがフォームから削除されます。

### ページナビゲーションコントロールを追加する

複数ページフォームを作成する際、ユーザに対し任意のページから別のページに移動するための手段を提供する必要があります。4Dでは3 種類の移動ツールを使用することができます:

- **タブコントロール**:タブコントロールオブジェクトを使用すると、個々のページへランダムにアクセスすることができます。タブコントロールはフォームの背景ページ上に配置し、そのプロパティを使用してページナビゲーションコントロールを提供します。
- **自動動作ボタン**:自動ページ移動ボタン(先頭ページ、最終ページ、前ページ、次ページ)をフォームへ追加することができます。これらのボタンは、背景ページ上に配置してください。
- **オブジェクトメソッド**:ランゲージにはFORM GOTO PAGEコマンドが用意されています。オブジェクトメソッドの一部としてこのコマンドを使用し、適切なオブジェクトタイプを用いて独自の移動制御を設置することができます。例えば、ピクチャボタンやポップアップメニューをページ移動機能として利用することも可能です。

**ページ移動ボタンを追加する**
フォームウィザードを用いてフォームを生成する際に、ボタンページでページ移動ボタンを組み込むことができます。フォームが生成されてから、そのフォームをフォームエディターで開き、必要なページを追加します。フォームの作成後にページ移動ボタンを追加する必要があれば、オブジェクトバーのボタン作成ツールを使用してボタンを追加します。詳細についてはボタンを参照してください。

**タブコントロールを使用する**
タブコントロールを使用すると、カレントページと他のページがひと目で分かります。タブコントロールの作成と使用に関する詳細はタブコントロールを参照してください。

# 継承フォームを使用する

4Dでは"継承フォーム"を使用することができます。これはフォームBのすべてのオブジェクトをフォームAで使用可能であるということです。つまりフォームAはフォームBからオブジェクトを"継承"します。

例えば、データベースに含まれるすべての入力フォームにOK、キャンセル、次レコード、前レコードボタン、およびロゴを配置したいとします。この場合、これらの要素だけが納められたフォームを作成し、このフォームを継承フォームとしてデータベースの全入力フォームから呼び出します。各入力フォームには、そのフォームの用途に限定されたフィールドとオブジェクトだけが含まれます。

フォームウィザードで定義されるフォームテンプレートとは異なり(フォームテンプレートを作成する参照)、継承フォームへの参照は常にアクティブです。そのため、継承フォームの要素が変更されると(例えば、ボタンスタイル)、この要素を使用する全フォームが自動的に変更されます。

テーブルフォームおよびプロジェクトフォーム両方とも継承フォームを使用する、あるいは継承フォームとして使用されることができます。

## 継承フォームを使用する

データベースを使用すると、継承フォームのオブジェクトと開かれたフォームのオブジェクトが動的に結び付けられます。このメカニズムは"ページ0"フォームメカニズムと非常に似ていますが、データベースの一連のフォームに使用される点が異なります。

フォームが実行されると、オブジェクトがロードされ、次の順序で結び付けられます:

1. 継承フォームの0ページ
2. 継承フォームの1 ページ
3. 開かれているフォームの0ページ
4. 開かれているフォームのカレントページ

この順序により、フォームにおけるオブジェクトの入力順が決まります。

**注:** 継承フォームの0 ページと1 ページだけが他のフォームに表示可能です。

継承フォームとして使用される場合、継承フォームのプロパティ(ウィンドウタイトル、サイズ調整、イベント等)とフォームメソッドは使用されません。他方、継承フォームに含まれるオブジェクトのメソッドは呼び出されます。

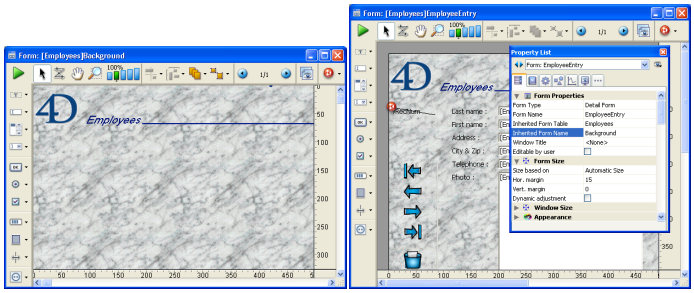
## 継承フォームを設定する

フォームの継承は、4D のフォームエディターで開始します。

継承フォームを設定するには:

1. 他のフォームを継承したいフォームをフォームエディターで開く。
2. プロパティリストを表示し、(オブジェクト以外の場所をクリックして) フォームプロパティを表示する。
   "継承されたフォームテーブル"と"継承されたフォーム名"リストが利用できます。すべてのデータベーステーブルとそのフォームが表示されます。
3. テーブルを選択し、次に継承するフォームを選択します。プロジェクトフォームを継承するには"継承されたフォームテーブル"で<なし>を選択します。

すべてのフォームを継承フォームとして指定できますが、そこに含まれる要素は、異なるデータベーステーブルで使用されても互換性が保たれるようになっていなければなりません。




継承フォームを選択すると、その内容が現在の編集ウィンドウに表示されます。これは単なるプレビューなので、このフォームのオブジェクトを選択したり、変更することはできません。選択や変更を行うには、別のウィンドウでその継承フォームを開かなければなりません。

フォームメニューやエディターのコンテキストメニューの、表示サブメニューで継承フォームオプションを選択/解除すると、継承フォームのオブジェクトを隠すことができます。

フォームの継承をやめるには、プロパティリストで**<なし>**オプションを選択します。

**注:** 任意のフォームで継承フォームを設定し、そのフォームを第3 のフォームの継承フォームとして使用することができます。再帰的な方法で各オブジェクトが連結されます。4Dは、再帰的ループが継承フォームとして定義されているかどうかを見つけ出し(例えば、[テーブル1]フォーム1 が[テーブル1]フォーム1 の継承フォームとして定義されている場合、つまり自分自身を指定している場合)、フォームの連鎖を中断します。

## データの入力順を変更する

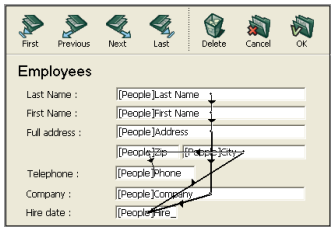
データ入力順とは、入力フォームでTab キーや**改行**キーを押したときに、フィールドやサブフォーム、その他のアクティブオブジェクトが選択される順番です。**Shift+Tab**や**Shift+改行**キーを押すことで逆方向に移動することもできます。

値が入力できるすべての入力可エリアはデータ入力順に含められます。(ラジオボタンやチェックボックスとして表示される) ブールフィールド、サブフォーム、コンボボックス、そしてピクチャーを受け入れるエリアもデータ入力順に含められます。

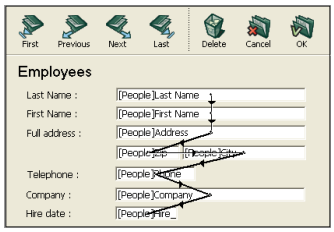
サーモメーター、ルーラー、ダイアルもデータを入力するために使用されます。しかしこれらのオブジェクトはデータ入力順には含まれません。これらのオブジェクトを操作するにはマウスでクリックします。

独自の入力順を指定しない場合、デフォルトとして4D はオブジェクトの階層に従い、"背面から前面"へ向けて入力順を決定します。したがって標準の入力順は、フォーム上でオブジェクトが作成された順番になります。

一部のフォームでは、独自の入力順が必要になります。例えば、次の図ではフォームの作成後に、住所に関連するフィールドが追加されています。この結果、標準の入力順が意味をなさなくなり、扱いづらい順番でデータを入力しなければなりません:




このようなケースでは、独自のデータ入力順を指定すると、より理にかなった順序でデータを入力できるようになります:




### データ入力順の表示と変更

入力順バッジまたは入力順モードを使用して、現在の入力順を表示することができます。しかし、入力順を変更するには、「入力順」モードを使用しなければなりません。

この節では入力順モードを用いて入力順の表示と変更を行う方法について説明します。バッジを用いた入力順表示についての詳細は[バッジを使用する](#)を参照してください。

入力順モードに切り替え、入力順を変更するには:

1. **フォーム**メニューの**入力順**コマンドを選択するか、またはツールバーの ボタンをクリックします。
   ポインタが入力順ポインタに変わり、4Dはフォーム上に線を引いて、データ入力時にオブジェクトが選択される順序を示します。
   ツールパレット上の他のツールをクリックするまでは、入力順序の表示と変更動作しか行えません。
2. データ入力順を変更するには、フォームのオブジェクト上にポインタを置き、マウスボタンを押したまま、次の入力順に設定したいオブジェクトまでポインタをドラッグする。

   これに応じて、4Dはデータ入力順を調整します。
3. 入力順を設定したい数だけ、ステップ2 を繰り返す。
4. 入力順を設定が終了したら、ツールバーの他のツールをクリックするか、または**フォーム**メニューから**入力順**を選択する。
   4Dは、フォームエディターの通常操作に戻ります。

**注:** フォームのカレントページの入力順だけが表示されます。フォームのページ0 や継承フォームに入力可オブジェクトが含まれている場合、デフォルトの入力順は次のようになります。継承フォームのページ0 のオブジェクト→ 継承フォームのページ1 のオブジェクト→ 開かれているフォームのページ0 のオブジェクト→ 開かれているフォームのカレントページのオブジェクト。

### データ入力順の先頭オブジェクトを設定する

すべての入力可オブジェクトは入力順に組み込まれます。入力順の1 番目のオブジェクトを定義するには、フォームレベルでそのオブジェクトの位置を変更しなければなりません。入力順モードは無効にしてください。

選択したオブジェクトを入力順序の1 番目に設定するには:

1. 入力順の1 番目にしたいオブジェクトを選択する。
2. **オブジェクト**メニューから**背面へ**を選択する。
   または
   オブジェクトのコンテキストメニューの**レベル**サブメニューから**背面へ移動**を選択する
3. **フォーム**メニューから**入力順**を選択するか、ツールバーの ボタンをクリックする。
   選択されたオブジェクトが入力順の先頭オブジェクトになり、以前の先頭オブジェクトは2番目になります。新しい先頭オブジェクトからオブジェクトへドラッグを行い、希望する入力順を設定することができます。
4. 設定が終了したら、再び**フォーム**メニューから**入力順**を選択するか、ツールバーの他のツールをクリックする。
   通常のフォームエディタに戻ります。

### データ入力グループを使用する

入力順序を変更する際に、フォームからオブジェクトグループを選択し、そのグループ内のオブジェクトに対して標準の入力順序を適用することも可能です。これにより、フィールドがグループや列に分かれているフォーム上で、データ入力順序を簡単に設定することができます。

データ入力グループを作成するには:

1. **フォーム**メニューから**入力順**を選択するか、またはツールバーの ボタンをクリックする。
2. データ入力用のグループに指定したいオブジェクトの周囲をマーキーで囲む。

マウスボタンを放すと、マーキーに囲まれているオブジェクトや、その矩形に接しているオブジェクトが標準入力順に設定されます。それ以外のオブジェクトのデータ入力順は、必要に応じて調整されます。

### フィールドを入力順から除外する

デフォルトでは、すべてのフィールドが入力順に組み込まれています。フィールドの入力可属性を保持したまま任意のフィールドを入力順から除外したい場合は、フィールドの**タブ有効**プロパティの選択を解除します。詳細は[データ入力の制御と補助](#)を参照してください。

# フォームオブジェクトの挿入と配置

## オブジェクトを追加する

フォームにオブジェクトを追加する方法は複数あります:

- オブジェクトバーでオブジェクトタイプを選択し、フォームエディター上で直接それを描画する (オブジェクトバーを使用する参照)。
- オブジェクトバーからオブジェクトをドラッグ＆ドロップする。
- 定義済みオブジェクトライブラリー (設定済みライブラリを使用する参照) またはカスタムライブラリー (カスタムオブジェクトライブラリの作成と使用参照) から選択したオブジェクトをドラッグ＆ドロップあるいはコピー/ペーストする。
- 他のフォームからオブジェクトをドラッグ＆ドロップあるいはコピー/ペーストする。
- エクスプローラー (フィールド) やデザインモードの他のエディター (リストやピクチャー等) からオブジェクトをドラッグ＆ドロップする。

オブジェクトをフォームに配置したら、フォームエディターを使用してそのオブジェクトのプロパティを編集できます。

フォームでは2つのタイプのオブジェクトを扱います:

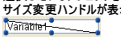
- **スタティックオブジェクト** (線、枠、背景ピクチャー等): これらのオブジェクトは通常フォームのアピアランスやラベル、さらにはグラフィックインターフェースを設定するために使用されます。これらはフォームエディターのオブジェクトバーから利用できます。プロパティリストを使用してこれらのグラフィック属性 (サイズ、カラー、フォント等) やリサイズオプションも指定できます。スタティックオブジェクトには、アクティブオブジェクトと異なり、変数が割り当てられません。しかしスタティックオブジェクトにダイナミックオブジェクトを挿入することは可能です (スタティックテキスト中で参照を使用する参照)。
- **アクティブオブジェクト**: この種のオブジェクトはインターフェース中でタスクや機能を実行します。フィールド、ボタン、スクロールリストなど様々な種類があります。各アクティブオブジェクトにはフィールドまたは変数が割り当てられます。アクティブオブジェクトについてはアクティブオブジェクトを使用するで説明します。

## オブジェクトを選択する

オブジェクトの操作（線幅やフォントを変更する）を行う前に、修正しようとするオブジェクトを選択する必要があります。

ツールバーを使用してオブジェクトを選択するには:

1. ツールバーの矢印ツール をクリックする。
   カーソルをフォームエリアに移動すると、カーソルは標準の矢印の形をしたポインターに変わります。
2. 選択したいオブジェクトをクリックする。
   サイズ変更ハンドルが表示され、オブジェクトが選択されたことを表わします。




プロパティリストを使用してオブジェクトを選択するには:

1. プロパティリストの一番上にあるオブジェクトリストドロップダウンリストからオブジェクト名を選択します。
   この方法では、他のオブジェクトの下に隠れているオブジェクトや、カレントウィンドウの表示領域外に置かれているオブジェクトを選択することができます。

オブジェクトの選択を解除するには、オブジェクト境界の外側をクリックするか、またはオブジェクト上で**Shift+クリック**します。

**注:** デザインモードを検索した結果ウィンドウでオブジェクトをダブルクリックして選択することもできます (名称変更参照)。

フォームウィザードで作成されたオブジェクトが選択できない場合、 ボタンをクリックするかページメニューから**0**を選択して背景ページに移動し、もう一度試してください。**Alt** (Windows) または **Option** (Mac OS) を押しながらクリックすることで直接0ページのオブジェクトにアクセスすることもできます。**フォーム**メニューの**表示**サブメニューまたはエディターのコンテキストメニューでページ0の選択を解除すると、背景ページ上のオブジェクトを非表示にできます。

### 複数のオブジェクトを選択する

複数のフォームオブジェクトに関して同じ操作を適用したい場合があります。例えば、オブジェクトの移動や整列、外観の変更を行う場合などです。4Dでは一度に複数のオブジェクトを選択することができます。複数のオブジェクトを選択する方法はいくつかあります:

- **編集**メニューから**すべてを選択**を選択して、すべてのオブジェクトを選択する。
- オブジェクト上でマウスの右ボタンをクリックし、コンテキストメニューから**同種のオブジェクトを選択**コマンドを選択する。
- **Shift** キーを押しながら、選択したいオブジェクトをクリックする。
- 選択したいオブジェクトグループの外側から各オブジェクトを囲むようにマーキー（選択矩形とも呼ばれます）を描画する。マウスボタンを離すと、マーキー内及びマーキーに重なるオブジェクトが選択されます。
- **Alt** キー（Windows）または、**Option** キー（Mac OS）を押しながら、マーキーを描画します。マーキーに完全に囲まれたオブジェクトが選択されます。

次の図はマーキーが描画され、2つのオブジェクトが選択されている様子を示しています:

一連の選択オブジェクトのなかから任意のオブジェクトの選択を取り消すには、**Shift**キーを押しながらそのオブジェクトをクリックします。この場合、他のオブジェクトは選択されたままになります。選択されているオブジェクトをすべて選択解除するには、
いずれのオブジェクトの境界にもかからない場所をクリックします。

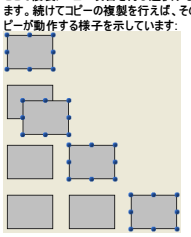
## オブジェクトを複製する

アクティブオブジェクトを含む任意のオブジェクトをフォーム上で複製できます。アクティブオブジェクトのコピーはオブジェクト名を除き、変数名、型、標準アクション、表示フォーマット、オブジェクトメソッドなどすべてのプロパティが保持されます。
ツールパレットの複製ツールを使用してオブジェクトを直接複製するか、行列を指定して複製ダイアログボックスでオブジェクトを複数一気に作成できます。このダイアログでは2つのコピー間の間隔も指定できます。

変数を複製する際は、複製するコピーの名前に自動で番号を振る特別な機能を使用することもできます。この点についてはグリッド上にオブジェクト作成を参照してください。

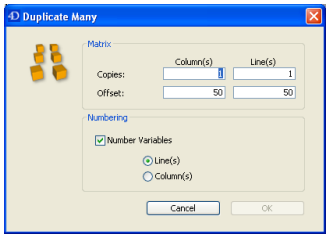
オブジェクトを複製するには:

1. 複製したいオブジェクトを選択する。
2. **編集**メニューから**複製**を選択する。
   4Dは選択されたオブジェクトのコピーを作成し、オリジナルオブジェクトの基点から右下に配置します。複製されたコピーはオリジナルの前面におかれます。
3. コピーを適切な場所に配置します。
   ここで複製メニュー項目を再び選択すると、4Dはもう一つコピーを作成し、最初のコピーと元のオブジェクトの間との同じ距離同じ方向に配置します。続けてコピーの複製を行えば、その前のオブジェクトと相対的に同じ距離同じ方向にコピーが作成されます。以下の図はこの相対的なコピーが動作する様子を示しています:



### 行列を指定して複製

"行列を指定して複製"ダイアログボックスは、**オブジェクト**メニューから**行列を指定して複製...**コマンドを選択すると表示されます。




- 上のエリアには作成したいオブジェクトの列数と行数を入力します。
  例えば3列2行のオブジェクトを作成したい場合、列に3を、行に2を入力します。横に3つの新しいコピーを作成したい場合は、列欄に4を入力

し、行はデフォルトの1のままにします。
- 列と行それぞれに、コピー毎のオフセットを指定できます。
  値はポイント単位で指定します。値は元のオブジェクトに対し相対的にコピー毎に適用されます。
  例えばオブジェクトごと縦方向にに20ポイントのオフセットを残し、元のオブジェクトの高さが50ポイントである場合、列のオフセットエリアに70を入力します。
- 格子状に変数を作成したい場合、**変数番号**オプションを選択し、番号を付ける方向を行または列から選択します。
  このオプションは選択したオブジェクトが変数の場合にのみ有効になります。詳細はグリッド上にオブジェクト作成を参照してください。

### オブジェクトを移動する

テンプレートで作成されたフィールドやオブジェクトを含め、フォーム上のグラフィックやアクティブオブジェクトはすべて移動可能です。オブジェクトを移動する場合は、次のような方法があります:

- オブジェクトをドラッグして移動する。
- 矢印キーを使用して、オブジェクトを一度に1 ピクセルずつ移動する。
- 矢印キーを使用して、オブジェクトを1ステップずつ移動する (デフォルトで1ステップ=20ピクセル)。
- プロパティリストにオブジェクトの座標を入力する (オブジェクトサイズを変更する参照)。

選択したオブジェクトのドラッグを開始すると、ハンドルが消えます。4Dはルーラーにオブジェクトの座標を示すマーカーを表示するので、適切な位置にオブジェクトを配置することができます。ハンドルをドラッグしないようにしてください。ハンドルをドラッグするとオブジェクトのサイズが変更されます。**Shift**キーを押しながらドラッグすると制約付きの移動になります。

マグネティックグリッドが有効な場合、グリッドに吸着するようにオブジェクトが移動されます。詳細はマグネティックグリッドを使用するを参照してください。

オブジェクトを1ピクセルずつ移動するには:

1. 移動したいオブジェクトを選択する。
2. キーボード上の矢印キーを使用して、オブジェクトを移動する。
   印キーを押すたびに、矢印の方向へオブジェクトが1 ピクセルずつ移動します。

オブジェクトを1ステップずつ移動するには:

1. 移動したいオブジェクトを選択する。
2. **Shift**キーを押しながらキーボード上の矢印キーを使用して、オブジェクトを移動する。
   デフォルトで1ステップで20ピクセル移動します。このピクセル数は環境設定のフォームページで変更できます。

### オブジェクトのグループ化

4Dではオブジェクトをグループ化して、そのグループをひとつのオブジェクトとして選択、移動、修正することができます。オブジェクトをグループ化した場合でも、互いに対する位置関係は変わりません。一般的には、フィールドとそのラベル、透明ボタンとそのアイコン等をグループ化するでしょう。あるグループが別のグループの一部となることもできます。

グループの大きさを変更すると、そのグループ内の全オブジェクトのサイズが同じ比率で変更されます (テキストエリアは除きます。テキストエリアのサイズは、そのフォントサイズに合わせて変更されます)。

ラジオボタンをグループ化すると、そのグループ中のラジオボタンはクリック時に排他的に選択されるようになります (ラジオボタンとピクチャーラジオボタン参照)。またグループ化はグループ化されたスクロールエリアを同期するためにも使用されます。

グループ化を解除すると、再び個々にオブジェクトを扱えるようになります。またグループ化したままでもオブジェクト上を**Ctrl+クリック** (Windows) または**Command+クリック**(Mac OS) して、オブジェクトのプロパティにアクセスすることができます (先にグループを選択しておく必要があります)。

グループ化はフォームエディター上でのみ意味を持ちます。フォームの実行中はグループ化されたすべてのオブジェクトが、グループ化されていないのと同じに動作します (ラジオボタンとスクロールエリアを除きます)。

**注:** 異なるビューに属するオブジェクトをグループ化することはできず、カレントビューに属するオブジェクトのみをグループ化することができます (オブジェクトビューを使用する参照)。

オブジェクトをグループ化するには:

1. グループ化したいオブジェクトを選択する。
2. **オブジェクト**メニューから**グループ化**を選択する。
   または
   フォームエディターのツールバーで**グループ化** ボタン をクリックする。
   4Dは、新たにグループ化されたオブジェクトの境界をハンドルで表わします。グループ内の各オブジェクトの境界にはハンドルが表示されません。
   これ以降、グループ化されたオブジェクトを修正すると、グループを構成する全オブジェクトが変更されます。

オブジェクトのグループ化を解除するには:

1. グループ化を解除したいグループオブジェクトを選択する。
2. **オブジェクト**メニューから**グループ解除**を選択する。
   または
   フォームエディタのツールバーで**グループ化解除**ボタンをクリックする。
   **グループ解除**が選択不可の場合、選択したオブジェクトは既に単独のオブジェクトに分けられていることを意味します。
   4Dは個々のオブジェクトの境界をハンドルで表わします。

### オブジェクトを整列する

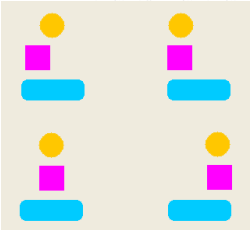
フォームエディタの整列ツールとグリッドツールを使用すると、フォーム上のオブジェクトを互いに整列させたり、または透明グリッドを用いて揃えることができます。

- あるオブジェクトを別のオブジェクトに揃える場合、オブジェクトの上端、下端、側面で揃えたり、または別のオブジェクトの縦や横の中心線に沿って揃えることができます。整列ツールを使用して、選択オブジェクトを直接揃えたり、または整列アシスタントを用いて、さらに詳細な整列を適用することも可能です。整列アシスタントを使用した場合、例えば、そのオブジェクトを位置の参照用として利用し、整列を適用する前にフォーム上での整列状態をプレビューできるようになります。
- 透明グリッドを使用すると、手動で各オブジェクトを別のオブジェクトに揃えることができます。その際、移動中のオブジェクトが別のオブジェクトに接近すると、"目に見える"位置ガイドとして点線が表示され、これに基づいて各オブジェクトの整列を実行することができます。

#### 即時整列ツールを使用する

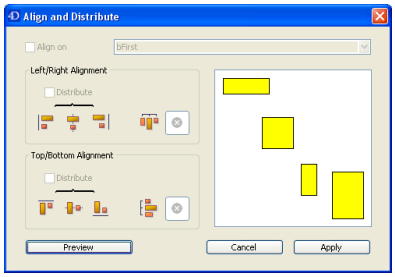
ツールバーの整列ツール (フォームエディター参照) または**オブジェクト**メニューの**整列** サブメニューを使用すると、選択したオブジェクトを瞬時に揃えることができます。

4D がオブジェクトを揃える場合、選択オブジェクトのうち1 つを定位置に置いたまま、残りのオブジェクトをそのオブジェクトに整列させます。この1 つのオブジェクトが"アンカー(基準オブジェクト)"となります。整列を行う方向で最も離れた位置にあるオブジェクトがアンカーとして使用され、他のオブジェクトはこのオブジェクトに合わせられます。例えば、一連のオブジェクトに対して右揃えを実行したい場合、一番右側に位置するオブジェクトがアンカーとして使用されます。
次の図は整列なし、左揃え、縦中央揃え、右揃えの状態を示しています:



#### 整列アシスタントを使用する

整列アシスタントを使用すると、オブジェクトに関するあらゆるタイプの整列や均等配置を実行することができます。




このダイアログボックスを表示するには揃えたいオブジェクトを選択し、**オブジェクト**メニューまたはエディタのコンテキストメニューの**整列**サブメニューから**整列...**コマンドを選択します。

1. "左/右整列"や"上/下整列"エリアで、実行しようとする整列に対応する整列アイコンをクリックする。
   見本エリアには、選択結果が表示されます。
2. 標準のアンカー方式による整列を実行するには、**プレビュー**または**適用**をクリックする。
   この場合、4Dは整列の方向に最も遠いオブジェクトをアンカーとして使用し、他のオブジェクトをそのオブジェクトに揃えます。例えば一連のオブジェクトに右揃えを適用すると、一番右のオブジェクトがアンカーとして使用されます。
   または:
   オブジェクトを特定のオブジェクトに揃えるには、**整列**オプションを選択し、さらに他のオブジェクトを揃えるための基準としたいオブジェクトをオブジェクト一覧から選択する。
   この場合、参照されたオブジェクトの位置は変わりません。

**プレビュー**ボタンをクリックすると、整列の結果をプレビューすることができます。するとフォームエディタ上のオブジェクトは整列しますが、ダイアログボックスが表示されたままなので、この整列のキャンセルや適用を行うことができます。

**注:** 整列アシスタントを使用すると、1 回の操作でオブジェクトの整列や均等配置を行えます。オブジェクトを均等配置する方法についての詳細はオブジェクトを均等配置するを参照してください。

#### マグネティックグリッドを使用する

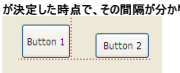
フォームエディタには仮想的なマグネティックグリッド機能があります。この機能は、フォーム上でオブジェクトの配置や整列を行う際に役立ちます。オブジェクトのマグネティック整列は、お互いに対するオブジェクトの位置に基づいて行われます。マグネティックグリッドは、少なくとも2 つのオブジェクトがフォーム上に存在する場合のみ使用可能です。

この機能は次のように作用します。フォーム上のあるオブジェクトを移動する際、4D は他のフォームオブジェクトとの相対的な位置関係に基づき、このオブジェクトの位置候補を示します。次のような場合には候補が示されます:

- 水平方向に2つのオブジェクトの端または中央が同じ位置になる場合。
- 垂直方向に2つのオブジェクトの端が同じ位置になる場合。

この状況になると、4Dはその位置にオブジェクトを配置し、そこが候補位置であることを示す赤いラインを表示します:

オブジェクトを均等配置する場合、4Dはインターフェース標準に基づいてオブジェクト間の距離を提示します。マグネティック整列と同様に、均等配置が決定した時点で、その間隔が分かりやすいように赤いラインで表わします。




この処理はあらゆるタイプのフォームオブジェクトに対して適用されます。**フォーム**メニューまたはエディタのコンテキストメニューの**マグネティックグリッド**コマンドを使用して、マグネティック機能をいつでも有効または無効に設定することができます。また、アプリケーション環境設定のフォームページにおいて、この機能をデフォルトで有効に設定しておくことも可能です。**Ctrl** キー（Windows）または**Control** キー（Mac OS）を押してオブジェクトを選択すると、手動でマグネティックグリッドが有効/無効に設定されます。

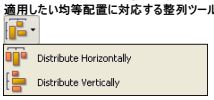
**注:** 手動でオブジェクトサイズを変更する場合も、このマグネティックグリッドの影響を受けます。

### オブジェクトを均等配置する

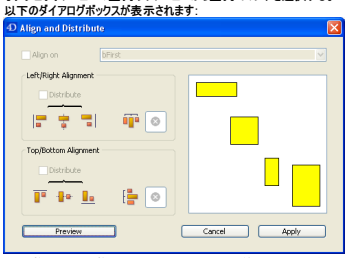
各オブジェクトが同じ間隔で配置されるように、オブジェクトを均等配置することができます。これを行うには、ツールパレットまたは整列アシスタントの均等配置ツールを用いてオブジェクトを均等に配置します。整列アシスタントを使用した場合は、1 回の操作でオブジェクトの整列と均等配置を実行できます。

**注:** マグネティックグリッドが有効である場合、オブジェクトを手動で動かすと、均等配置のためガイドが表示されます。詳細は前段落を参照してください。

同じ間隔を空けてオブジェクトを配置するには:

1. 3つ以上のオブジェクトを選択し、希望する整列ツールをクリックする。
2. 適用したい均等配置に対応する整列ツールをツールバー上でクリックする。
   
   
   または
   **オブジェクト**メニューまたはエディターのコンテキストメニューの**整列**サブメニューから均等配置用のメニューコマンドを選択する。
   4D は各オブジェクトを均等に配置します。各オブジェクトの中心までの間隔、および隣接する2 つのオブジェクト間にある間隔のうち最も広い間隔が基準として用いられます。

整列＆均等配置ダイアログボックスを用いてオブジェクトを均等に配置するには:

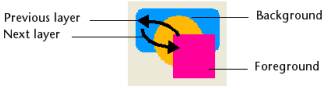
1. 均等配置したいオブジェクトを選択する。
2. **オブジェクト**メニューの**整列**サブメニューから**整列**コマンドを選択する。
   以下のダイアログボックスが表示されます:
   
   
3. 左/右整列や上/下整列エリアにおいて、標準の均等配置アイコンをクリックする。
   *(標準の横均等揃えアイコン)*
   見本エリアには、選択結果が表示されます。
4. 標準方式による均等配置を実行するには、**プレビュー**または**適用**をクリックする。
   この場合、4Dは標準の均等配置を実行し、オブジェクトは等間隔で配置されます。
   または:
   特定の均等配置を実行するには、均等配置オプションを選択する（例えば各オブジェクトの右側までの距離をもとにしてオブジェクトを均等に配置したい場合）。このオプションはスイッチのように機能します。**均等配置**チェックボックスが選択されていると、このオプションの下にあるアイコンは異なる動作を行います:
   - 左/右整列の場合、各アイコンは次の均等配置に対応します。選択オブジェクトの左側、中央（横）、右側で均等に揃えます。
   - 上/下整列の場合、各アイコンは次の均等配置に対応します。選択オブジェクトの上端、中央（縦）、下端で均等に揃えます。

   **プレビュー**ボタンをクリックすると、この設定による実際の結果をプレビューすることができます。ただし、この処理はフォームエディター上で実行されますが、ダイアログボックスは前面に表示されたままです。したがって、変更の**キャンセル**や**適用**を行うことができます。

**注:** このダイアログボックスでは、整列と均等配置を合わせて実行することができます。整列に関する詳細はオブジェクトを整列するを参照してください。

### オブジェクトを重ねる

フォーム上で他のオブジェクトを隠しているオブジェクトに関し、その位置を変える必要が生じることもあります。例えば、フォーム上のフィールドの背景にグラフィックを配置したい場合です。4D には**背面へ**、**前面へ**、**レベルを1 つ上げる**、**レベルを1 つ下げる**という4つのメニュー項目があり、これらのコマンドを使用してフォーム上のオブジェクトを"重ねる（レイヤー）"ことができます。この重なりはデフォルトの入力順も規定します (データの入力順を変更する参照)。次の図は、他のオブジェクトの前面に置かれたオブジェクトと、他のオブジェクトの背面に置かれたオブジェクトを示しています:




オブジェクトレベルを変更するには、以下のいずれかの作業を行います:

- **オブジェクト**メニューから**背面へ**、**前面へ**、**レベルを1 つ上げる**、**レベルを1 つ下げる**のいずれかを選択する
- エディターのコンテキストメニューの**レベル>** サブメニュー内のコマンドのいずれかを選択する
- ツールバーのレベル管理ボタンに割り当てられたコマンドのいずれかを選択する

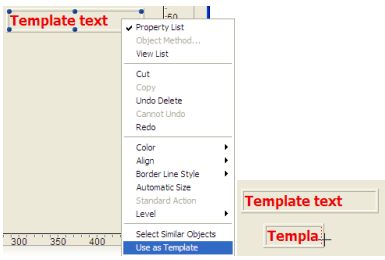
**注:** 複数のオブジェクトが重なっている時、**Ctrl+クリック** / **Command+クリック**ショートカットを使用して、クリックするたびに下のレイヤーにあるオブジェクトを選択できます。

異なるレベルを並び替える際、4Dは常に後ろから前に進みます。結果として、*以前の*レベルは選択されたオブジェクトを1レベル後ろに向けて移動します。*次の*レベルは選択されたオブジェクトを1レベル前に向けて移動します。

**オブジェクトテンプレートを設定する**

**テンプレートとして使用**コマンドは、オブジェクトが選択されたときにコンテキストメニューから利用可能となり、クリックされたオブジェクトをテンプレートとして保存します。このテンプレートとそのプロパティは、同種のオブジェクトを作成する際に使用されます。

以下の例で、このコマンドはテキストオブジェクトに適用されます。以降作成されるすべてのテキストオブジェクトはデフォルトで、指定したオブジェクトのプロパティを使用します:




オブジェクトタイプごとにカスタマイズされたテンプレートを使用できます。データベース全体で、オブジェクトごとに1つのテンプレートを持つことができます。

テンプレートには、コマンドが選択された時のすべてのオブジェクトプロパティが保存されています (固定座標とオブジェクトメソッドを除く)。

**オブジェクトテンプレートを削除する**

オブジェクトテンプレートを更新したり置き換えたりする場合は、同じオブジェクトタイプで新しいテンプレートを作成します。

フォームエディターの**オブジェクト**メニューから**カスタムテンプレートをクリア**コマンドを使用して、すべてのオブジェクトテンプレートを削除し、標準のテンプレートに戻すことができます。このコマンドを選択すると確認ダイアログが表示されます。このダイアログボックスを受け入れると、すべてのオブジェクトテンプレートがデフォルト値にリセットされます。

## オブジェクトの表示プロパティを設定する

フォーム中のオブジェクトはそれぞれプロパティを持ち、アプリケーションモードでのアピアランスや表示方法、リサイズ方法などが指定されます。これらのプロパティはプロパティリストからアクセスします (フォームエディター参照)。

ダイナミックオブジェクトに関する特定のプロパティはアクティブオブジェクトを使用する、リストボックス、そしてサブフォームとウィジェットで説明します。

### オブジェクトのサイズ

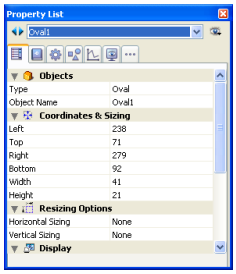
フォームエディターでオブジェクトのサイズを設定する方法には以下があります:

- サイズ変更ハンドルをドラッグする。
- Ctrl (Windows) やCommand (Mac OS) と矢印キーを使用して1ピクセルずつ変更する。
- ステップごとに変更する (デフォルトで1ステップ=20ピクセル)。
- プロパティリストに値を入力する。

ハンドルを使用する標準の方法でオブジェクトのサイズを変更できます。**Shift**キーを押してハンドルをドラッグすると制約付きのサイズ変更となります。線は水平、45°、あるいは縦にひかれ、四角は正方形に、楕円は円になります。マグネティックグリッドが有効になっていると、手作業によるサイズ変更は5ピクセル単位になります。マグネティックグリッドについてはマグネティックグリッドを使用するを参照してください。

1ピクセルずつオブジェクトをサイズ変更するには、オブジェクトを選択し、**Ctrl** (Windows) または **Command** (Mac OS) キーを押しながら矢印キーを使用します。上または下矢印キーを押すとオブジェクトの高さが変更され、左あるいは右矢印キーを押すとオブジェクトの幅が変更されます。同時に**Shift**キーを押すとステップ単位 (デフォルトで20ピクセル) でサイズ変更されます。デフォルトのステップ単位は環境設定のフォームページで変更できます。

座標を入力してオブジェクトを移動したりサイズ変更したりするには、オブジェクトを選択して、プロパティリストの"位置調整＆サイズ調整"テーマを展開します。座標はルーラーに設定した単位に基づきピクセル、センチ、インチ単位で表示されます。フォームエリアの左上座標が0,0です。座標入力エリアに新しい座標を入力します。




4Dは入力された座標にオブジェクトを移動します。指定した値に基づき、オブジェクトはサイズ変更や移動されます。

**注:** マルチプラットフォームのコンテキストでは、スタイルシートを適用することでオブジェクトの高さが可変になることがあります。オブジェクトの高さはカレントのプラットフォームに設定されたフォントの高さに基づき倍数で計算されるためです。スタイルシートについてはスタイルシートを参照してください。

#### 自動サイズ

スタティックテキストエリアとピクチャーを、現在のグラフィック特性 (フォントサイズ、スタイル等) に基づき自動で最適表示サイズにすることができます:

1. オブジェクトを右クリックし、コンテキストメニューから**自動サイズ**を選択する。
   または
   **Ctrl** (Windows) または **Command** (Mac OS) キーを押しながらオブジェクトの右下のハンドルをクリックする。

このコマンドは以下の効果を及ぼします:

- スタティックテキストエリア、チェックボックス、ラジオボタン、そしてボタンは含まれるテキスト (ラベル) 丁度の大きさに調整されます。
- フィールドと変数は最低高さが選択されているフォントサイズに、幅が100ポイントになります。オブジェクトサイズがこれらより大きい場合、変更はされません。
- ピクチャーやピクチャーボタンはソースピクチャーを元のサイズで表示する大きさになります。
- スクロールエリアと階層リストはカレントフォントに基づいて高さが調整されます。
- コンボボックス、ポップアップ/ドロップダウンリスト、階層ポップアップメニューは、現在の高さが小さい場合、デフォルトの最低の高さで表示されます。

### リサイズ

アプリケーションモードでユーザーがフォームを表示しているウィンドウのサイズを変更すると、そこに含まれるオブジェクトもサイズ変更されたり、移動されたりします。

フォームを表示するウィンドウのサイズをユーザが変更すると、自動サイズ調整が行われます。自動サイズ調整により、フォームを拡げるとオブジェクトも拡大されます (または拡げたウィンドウが縮小された場合にはオブジェクトも縮小されます)。例えば入力フォーム上でフィールドを囲む四角形を使用している場合は、ユーザがウィンドウを拡大すると、自動サイズ調整によりその四角形はウィンドウの端まで拡げられます。

また、自動位置調整機能を利用することも可能です。フォームサイズが変更されると、自動位置調整によりオブジェクトは水平方向または垂直方向に移動されます。自動位置調整が有効である場合、ユーザがウィンドウサイズを小さくすると、4D は見える位置にそのオブジェクトを維持しようとします。例えば、ボタンの列のサイズが変更されて、一部のボタンが隠れてしまった場合、自動位置調整により、ボタンは水平方向または垂直方向へ移動され、ボタンが表示される位置に調整されます。

自動サイズ調整機能や自動位置調整機能は、プロパティリストのリサイズオプションテーマで有効にすることができます。このテーマには水平サイジングと垂直サイジングという2 つの行があり、3 種類のプロパティを指定することができます (**なし**、**拡大**、**移動**)。

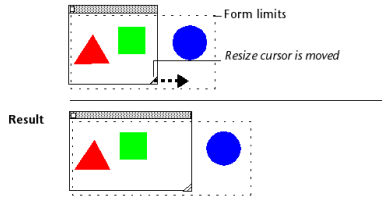
| オプション | 結果 |
|---|---|
| 水平方向サイジング: 拡大 | ユーザがウィンドウの幅を変更すると、オブジェクトの幅にも同じ割合を適用します。 |
| 水平方向サイジング: 移動 | ユーザがウィンドウの幅を変更すると、幅の増加分と同じだけオブジェクトを左か右に移動します。 |
| 垂直方向サイジング: 拡大 | ユーザがウィンドウの高さを変更すると、オブジェクトの高さにも同じ割合を適用します。 |
| 垂直方向サイジング: 移動 | ユーザがウィンドウの高さを変更すると、高さの変更分と同じだけオブジェクトを上か下に移動します。 |

位置調整オプションを使用すると、表示された状態を維持するために、オブジェクトは指定された方向へ移動されます。

**なし**オプションを指定すると、フォームサイズが変更されても、オブジェクトは固定されたままになります。

#### サイドプッシャー

4Dでは、デフォルトで各ウィンドウの右側面と下側面は"プッシャー"スプリッターとして機能します。つまりウィンドウを大きくすると、画面上のウィンドウ境界の右側や下側にあるオブジェクトが右方向や下方向へ自動的に押し出されます:




**注:** この機能はスクロールバー付きのウィンドウでは動作しません。

### デフォルトで非表示

大部分のオブジェクトに対して**デフォルトで非表示**プロパティを指定することができます。

プロパティリストの"表示"テーマにあるこのプロパティを使用すると、ダイナミックなインターフェースを容易に開発できるようになります。このようなインターフェースを開発するとき、多くの場合はフォームのOn Loadイベント中にプログラムからオブジェクトを非表示にした後で、一部のオブジェクトを再度表示する必要性が頻繁に生じます。

**デフォルトで非表示**プロパティを使用すると、特定オブジェクトをデフォルトで非表示にしておくことにより、このロジックを逆に働かせることができます。この後、開発者は状況に応じてOBJECT SET VISIBLEコマンドを使用し、これらのオブジェクトを表示するようプログラミングすることができます。このコマンドに関する詳細は4Dランゲージリファレンスマニュアルを参照してください。

**注:** このプロパティとポップアップメニューの非表示プロパティを混同しないでください。ポップアップメニューの非表示プロパティでも、オブジェクトが表示されなくなりますが、そのオブジェクトはアクティブな状態のままです。

### プラットフォーム

プロパティリストの"アピアランス"テーマでオブジェクトごとにアピアランスを設定できます。これを行うには以下の選択肢があります:

- **フォームから継承**: オブジェクトのプラットフォームインターフェースはフォームのそれと同じです。フォームのプラットフォームインターフェースはフォームプロパティで設定されます。
- **システム**: フォームレベルに設定されたプラットフォームにかかわらず、オブジェクトはアプリケーションが実行されているプラットフォームに従って描画されます。
- **印刷**: フォームレベルに設定されたプラットフォームにかかわらず、オブジェクトは印刷用に描画されます。

**注:** 以前のバージョンの4Dから変換されたデータベースの場合、追加のプラットフォームプロパティを選択できることがあります。プラットフォームインターフェースプロパティについてはフォームプロパティを参照してください。

**背景と境界線**

背景と境界プロパティを使用してオブジェクトの境界線やグラフィックアピアランスを設定できます。

**透過**

**透過**属性は、透明または不透明になることができるオブジェクト (フィールド、変数、リストボックス、ボタン、ピクチャー、スタティックテキスト等) の"背景色と境界線"テーマで利用できます。オブジェクトの背景を透明にして、フォームの背景色を使用することができます。

以下の例は透明属性を設定していない、また設定したスタティックピクチャーです:

**境界線スタイル**

プロパティリストの"背景色と境界線"やコンテキストメニューの**境界線スタイル**サブメニューを使用して、ほとんどのオブジェクトの境界線スタイルを変更できます。これを行うには以下の選択肢があります:

- なし: 境界線なし
- 標準: 連続した1ptの境界線
- ドット: 1ptの非連続境界線
- 浮き上がり: 3D効果 (浮き上がり) の境界線
- くぼみ: 3D効果 (くぼみ) の境界線
- 二重: 二重境界線

**注:** このテーマの他のプロパティは設定された線のタイプによって変化します。例えば"ドット"を選択すると、プロパティリストに**点線タイプ**プロパティが表示されます。

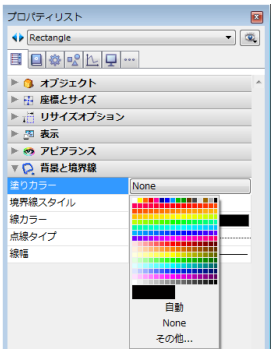
以下はWindowsにおける境界線スタイルの例です:

異なるフォームオブジェクトごとのこのオプションの効果については、ボタンとチェックボックスを参照してください。

**背景および枠線の色**

4Dではモニターへの表示や、カラープリンターでの印刷時に使用できる、カラープロパティをオブジェクトに設定できます。

コンテキストメニューのカラーコマンドまたはプロパティリストから直接選択して、背景と境界線に異なるカラーを設定できます。




**注:** プロパティリストでは背景色が**塗りカラー**、境界線の色が**線カラー**と記載されています。コンテキストメニューでは描画色が線カラーに対応します。

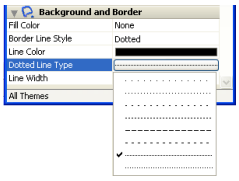
背景色ではプロパティリストで2つの標準オプションを利用できます:

- **自動**: オブジェクトはOSに設定された自動カラーを使用して表示されます。
- **なし**: オブジェクトの背景は透過になります。

プロパティリストでは**他...**を選択することでシステムカラーパレットを表示し、カスタム色を選択することができます。

**点線タイプ**

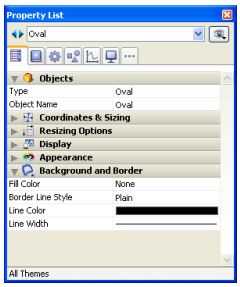
このプロパティは**境界線スタイル**で"ドット"を選択すると表示されます。ドット線フォーマットを選択するために使用します:




**線幅**

4Dでは線や、楕円、グリッド、四角などのオブジェクトに線幅を設定できます。

線幅はプロパティリストや**オブジェクト**メニューの**線幅**メニュー項目から指定します:




線幅を一つ選択します。プリンターで印刷できる最細幅である極細線及び1, 2, 3 あるいは 4ポイントの線幅から選択できます。**その他...**を選択すると、4Dは20ポイントまでを指定できるダイアログボックスを表示します。

**テキスト属性**

プロパティリストの"テキスト"テーマ、または**オブジェクト**メニューの**フォント**、**スタイル**、**カラー**サブメニューを使用して、テキストオブジェクトのフォント属性・

スタイル属性を設定できます。

以下の属性を設定できます:

- **スタイルシート** (プロパティリストのみ): オブジェクトにスタイルシートを割り当てる。スタイルシートについてはスタイルシートを参照してください。
- **フォント**: オブジェクトで使用するフォント名。
- **スタイル**: オブジェクトで使用するスタイル (ボールド、イタリック、下線)。
- **フォントサイズ**: ポイント単位のフォントサイズ。
- **フォントカラー**: オブジェクト中で使用されるフォントのカラー。コンテキストメニューでは描画色が対応します。
- **行スタイル配列/行フォントカラー配列**: スタイルとカラーを管理する配列名(配列型のリストボックスのみ)
- **横揃え**: エリア中のテキストの横位置
- **縦揃え**: エリア中のテキストの縦位置
- **回転**: テキストエリアの角度(回転)を修正します。これについての詳細な情報は、テキストの回転を参照して下さい。
- **マルチスタイル**: アプリケーションモードで、フィールドや変数などのテキストエリアにカスタムスタイルを適用できるようにします。詳細はマルチスタイル (リッチテキストエリア)を参照してください。
- **ピッカーの使用を許可**: OPEN FONT PICKER または OPEN COLOR PICKER を使用してユーザーがフォントまたはカラーピッカーが呼び出し、カレントセッションの間オブジェクトのフォントまたはカラーを変更することを許可します。このオプションはシステムピッカー(*font picker*と *color picker*)コマンドのアクションのスコープを管理します。フィールド、変数、そしてコンボボックス型のフォームオブジェクトにおいて使用することが出来ます。デフォルトでは、このオプションは全てのフォームオブジェクトにおいてチェックが外されています。システムピッカーウィンドウを使用してフォント・カラー属性を変更可能にしたいオブジェクトに関しては、全て明示的にこのオプションをチェックしなければなりません。

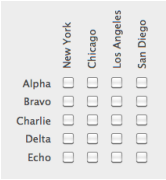
#### テキストレンダリングに関する互換性の注意 (OS X)

Appleからの推奨に従い、OS Xの4D v13ではテキストレンダリングに使用されている *frameworks* が更新され統一されています(廃止予定であったQD/MLTE *frameworks* を放棄し、CoreText の使用へと移行しました)。

この新しいCoreText *framework* への移行に伴う問題が最小限にとどまるように努力はしておりますが、OS Xの4D v13へと変換されたアプリケーションのテキストエリアのレンダリング(特に改行に関するレンダリング)において多少の変化を引き起こす可能性があります。この場合、変換する前のv12のデータベース内でいくつかのフォームオブジェクトをリサイズする必要があるかもしれません。

## テキストの回転

4D では、フォーム内のテキストエリアを回転させることができます:




テキストの回転機能は、フォームエディターから設定するか(恒久的なプロパティ)、OBJECT SET TEXT ORIENTATION コマンドを使用することによって設定することができます(カレントのプロセスの間有効)。

### 回転可能なオブジェクト

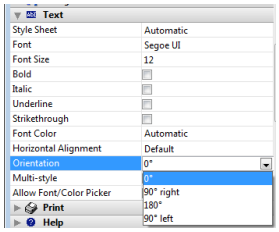
テキストの回転は、フォーム内の**入力不可**のテキストエリアにおいてのみ可能です:

- 静的なテキスト
- 入力不可のテキスト変数またはフィールド - 具体的には、テキストベースのオブジェクトで、文字列、日付、時間、数字、複数行、またはマルチスタイル型オブジェクトを含むオブジェクトをさします。

他の型のオブジェクト(ボタン、入力エリア、リスト、ラジオボタン等)は回転させることができません。

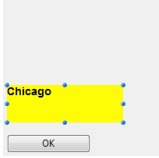
### 回転の設定

新しい"方向"のプロパティは、回転がサポートされているオブジェクトのプロパティリストのテキストのテーマ内にあります。テキストエリアは、90°単位で回転させることができます:

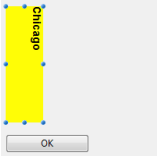



それぞれの回転角度を適用するとき、オブジェクトの左下の角は固定されたままで回転していきます:

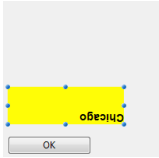
**0° (デフォルト):**



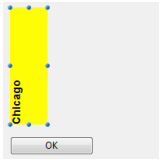

**90° 時計回り:**




**180°:**




**90° 反時計回り:**

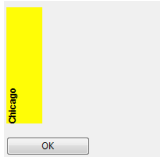



#### フォームエディターまたはランゲージコマンドを使用した回転

フォームエディター内で回転を適用したとき、テキストを含んでいるオブジェクトはそのテキストの回転に従って回転します。

ただし、OBJECT SET TEXT ORIENTATION コマンドを使用してテキストを回転させた場合にはこの原理はあてはまりません。このコマンドが実行されたとき、テキストは回転しますが、テキストを含むオブジェクトは回転しません。例えば、90° Left(90°反時計回り)をフォームエディターの"Chicago"というテキストに対してOBJECT SET TEXT ORIENTATIONコマンドを使用したときには、プロパティリストで設定した場合と比較して、フォームを実行した際に異なる結果が得られます:

**90° 反時計回り(プロパティリスト)** **90° 反時計回り(ランゲージ)**

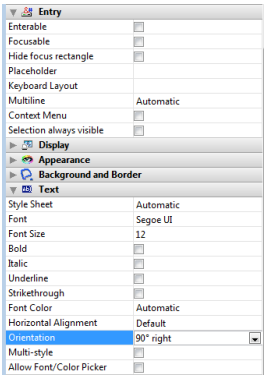


詳細な情報に関しては、OBJECT SET TEXT ORIENTATION コマンドの詳細を参照して下さい。

### フィールドと変数の回転

入力不可かつフォーカス不可のテキストオブジェクトに限り、回転させることが出来ます。

フィールドまたは変数型のオブジェクトに対して、方向のプロパティにて0°以外のオプションを選んだ場合、入力可 とフォーカス表示のプロパティは(選択されていた場合)自動的に解除されます:



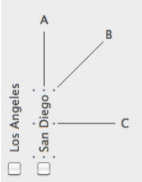

その際、このオブジェクトは自動的に入力順から除外され、背景が透明になります。

また逆に、回転しているオブジェクトに対して入力可もしくはフォーカス可のプロパティにチェックをした場合、方向のプロパティは自動的に0°へとリセットされます。

**回転しているオブジェクトの変更**

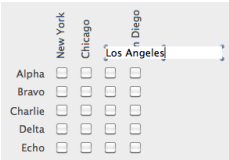
テキストが変更された後でも、そのセレクションハンドルや、OBJECT SET COORDINATESといったランゲージコマンド、またはプロパティなどを通じて、エリアのサイズや位置などを変更することは可能です。

テキストエリアの高さと幅は、回転の方向に依らないという点に注意して下さい:




- オブジェクトが A 方向にリサイズされるとき、変更されるのは幅(縦方向)です。
- オブジェクトが C 方向にリサイズされるとき、変更されるのは高さ(横方向)です。
- オブジェクトが B 方向にリサイズされるとき、幅と高さ両方が同時に変更されます。

エリアの中身は、フォームエディターを使用して修正することもできます。エディットモードに切り替わる際、テキストはデフォルトの方向へと戻ります:



## スタティックピクチャーを使用する

フォームでは2タイプのスタティックピクチャーを使用できます。フォームに直接格納される個別のピクチャーと、フォーム外に格納されたピクチャーへの参照です。ピクチャーが異なる場所で複数回使用される場合、最適化のために、参照として挿入されたピクチャーを使用することをお勧めします。こうすれば必要なピクチャーの実体は1つですみます。さらにピクチャーとその参照の間にはリンクがあり、ソースピクチャーを更新すれば、すべてのオカレンスも自動で更新されます。

4Dでは、参照として挿入されたピクチャーは"ライブラリーピクチャー"タイプで、直接格納されたピクチャーは"スタティックピクチャー"タイプです (プロパティリストのタイプフィールドに表示されます)。参照としてピクチャーを挿入するには、ピクチャーライブラリーまたはデータベースの**Resources**フォルダーからピクチャーをドラッグ＆ドロップします。

複数ページの0ページにピクチャーを配置すると、そのピクチャーは自動ですべてのページに背景として表示されます。またピクチャーを継承フォームに置くこともでき、そのフォームを継承するすべてのフォームの背景とすることもできます。どちらの方法も、それぞれのページにピクチャーを配置するより、アプリケーションの動作が速くなります。

フォームにおかれたスタティックピクチャーにはプロパティを割り当てることができます。例えば小さなピクチャーをフォームの背景として使用するために繰り返し表示プロパティを設定できます。ピクチャーはオブジェクト内で必要なだけ繰り返して表示されます。

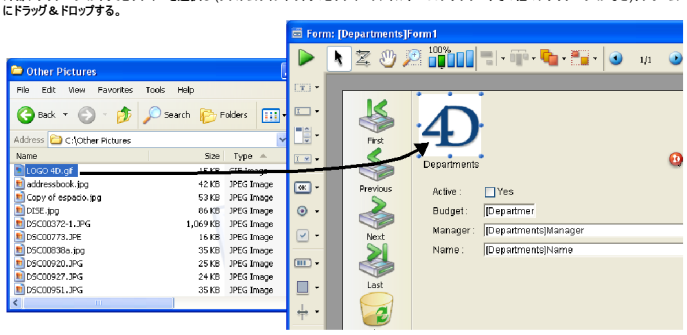
### スタティックピクチャーを挿入する

フォームにスタティックピクチャーを挿入する方法は3つあります。クリップボードからピクチャーを貼り付ける、ピクチャーファイルをドラッグ＆ドロップする、またはピクチャーライブラリーからピクチャーをドラッグ＆ドロップする、です。

#### コピー/ペーストやドラッグ＆ドロップで挿入する

スタティックピクチャーをフォームにペーストあるいはドロップするには:

1. ペーストボードにピクチャーをコピーし、フォームにペーストする。
   または
   外部アプリケーション内でピクチャーを選択し (システムウィンドウ内のピクチャーファイル、Webブラウザー、その他のアプリケーションなど)、フォームにドラッグ＆ドロップする。




   元のピクチャーは4Dがネイティブに管理するフォーマットでなければなりません (4Dは主なピクチャーフォーマットを認識します: JPEG, PNG, BMP, SVG, GIF, 等)。ピクチャーはネイティブフォーマットでターゲットエリアにペーストされます。
   **注**:
   - これはライブラリーピクチャー、オブジェクトに割り当てられたピクチャー (例えば3Dピクチャーボタン)、そしてアプリケーションモードのフィールドや変数でも動作します。
   - データベースの**Resources**フォルダーに格納されたファイルをドラッグ＆ドロップすると、ピクチャー参照を挿入できます ("ピクチャーファイルの自動参照"を参照)。
2. 必要に応じてピクチャーの位置とプロパティを調整する。

#### ピクチャーライブラリーを使用して挿入する

ピクチャーライブラリーの使用方法については[ピクチャーライブラリ]を参照してください。

ピクチャーライブラリーからフォームにピクチャーを配置するには:

1. ツールボックスのピクチャーライブラリーを開き、配置したいピクチャーを選択する。
2. ピクチャーをピクチャーライブラリーからフォームにドラッグする。
   - ドラッグしたピクチャーがフレームに分割されている場合、自動でピクチャーボタンとして挿入されます (ドラッグ時に**Shift**キーを押していた場合はピクチャーポップアップメニュー)。画像をスタティックピクチャーやライブラリーピクチャーとして挿入したい場合は、ドラッグ時に**Alt** (Windows) または**Option** (Mac OS) キーを押します。
   - 挿入されるピクチャーはデフォルトで**ライブラリーピクチャー**タイプです。4Dは参照を挿入し、自動でライブラリーピクチャーとのリンクを保守します。このリンクを解除したい場合、プロパティリストでピクチャーのタイプを**スタティックピクチャー**に変更します。スタティックピクチャーとライブラリーピクチャーのプロパティは通常一緒です。
3. ピクチャーの位置を調整し、プロパティを設定します。

ピクチャーは他のオブジェクト同様一連のプロパティを持ちます。必要に応じてこれらのプロパティを変更できます。

### ライブラリーソースとピクチャーのリンクを解除する

ピクチャーライブラリーのピクチャーを挿入すると、それはピクチャーへの参照を挿入したことになります。プロパティリストにはタイプとしてライブラリーピクチャーが表示されます。ピクチャーライブラリーでピクチャーが変更されると、そのインスタンスも更新されます。

ピクチャーライブラリーと、フォームに挿入されたピクチャー間のリンクを解除するには:

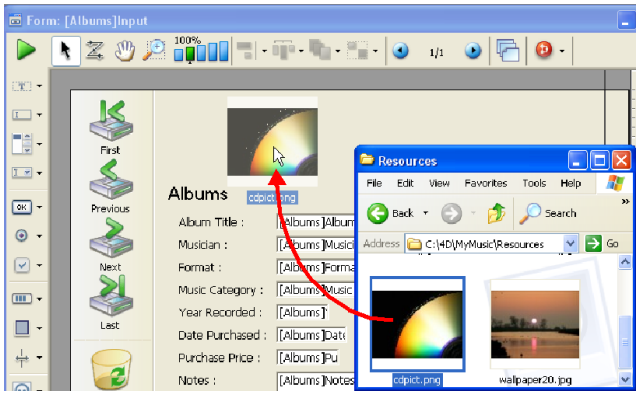
1. プロパティリストにピクチャーのプロパティを表示する。
2. "オブジェクト"テーマで**タイプ**行をクリックする。
   **ライブラリーピクチャー** (デフォルト値) と**スタティックピクチャー**、2つのタイプが選択可能です。
3. **スタティックピクチャー**を選択する。
   ピクチャーの参照が解除され、ペーストボードやファイルがドロップされたのと同様に扱われます。

### ピクチャーファイルの自動参照

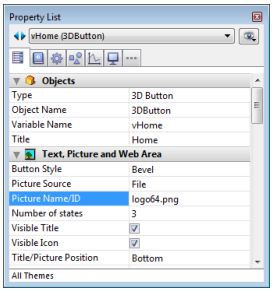
フォームで使用するスタティックピクチャーを格納し、参照を使用して扱うために、**Resources**フォルダーを使用できます。これらのピクチャーの表示は最適化され、管理がとても容易になります。

**注: Resources**フォルダーについての詳細は[4Dファイルの説明]を参照してください。

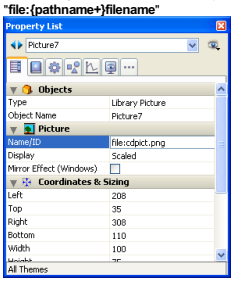
特に.png (ビットマップ) や.svg (ベクター) ピクチャーを使用することができます。これらのピクチャーのフォームへの配置は、**Resources**フォルダーからのドラッグ＆ドロップで行うことができます:




- ピクチャーボタンや3Dボタン、ピクチャーポップアップメニューなどのダイナミックオブジェクトにピクチャーをドロップした場合、4Dは自動でピクチャーソースを"ファイル"に設定し、また以下の形式でピクチャー参照を挿入します: "**{pathname+}filename**"




- フォームの背景上にピクチャーをドロップした場合、ライブラリピクチャーが作成され、その参照が以下の形式で配置されます:
  **"file:{pathname+}filename"**




ピクチャーを**Resources**フォルダーのサブフォルダーに配置することもできます。特に.lprojフォルダーメカニズムを使用して、異なる言語で異なるピクチャーを使用できます (このメカニズムに関する詳細は付録 C: XLIFFアーキテクチャー参照)。

**スタティックびkくちゃーの表示モードを設定する**

フォームに配置されたスタティックおよびライブラリーピクチャーの表示モードを設定できます。表示モードは、ピクチャーを含むエリアがリサイズされたときの、ピクチャーの再描画の方法を決定します。

これを行うにはピクチャーを選択し、プロパティリストの"ピクチャー"テーマから**表示**メニューを使用します:

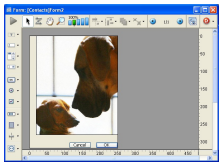
- **スケーリング** (デフォルトモード): ピクチャーオブジェクトがリサイズされると、ピクチャーもリサイズされ、ピクチャー全体が表示されたままになります。
- **切り捨て**: ピクチャーオブジェクトがリサイズされると、ピクチャーはその大きさを保持し、境界線だけが変更されます。ピクチャーは常にピクチャーオブジェクトの中央に表示されます。ピクチャーオブジェクトがピクチャーよりも小さくなると、ピクチャー表示は切り捨てられます。
- **繰り返し**:ピクチャーオブジェクトのサイズが大きくなると、新しいエリアを埋めるのに必要なだけピクチャーが繰り返されます。メモリー節約のため、背景画像の場合に使用することが推奨されます。ピクチャーオブジェクトのサイズがピクチャーサイズよりも小さくなると、表示は切り捨てられます (中央合わせはされません)。

**ミラー効果 (Windows)**

**ミラー効果 (Windows)** オプションはフォームに表示されるピクチャーのプロパティリストの"ピクチャー"テーマにあります。

このオプションは"right-to-left"言語に設定されたアプリケーションで使用でき、反転されたバージョンのインターフェースを持ちます ("right-to-left"モードに関する詳細は参照)。

デフォルトでright-to-leftモードはフォーム上のピクチャーを反転させません。しかし特定のピクチャーを反転させる必要が生じる場合があります。例えばオブジェクトの位置にリンクした矢印などです。ミラー効果 (Windows) オプションにチェックされていると、left-to-right (ミラー効果) 反転がピクチャーに適用されます:



このオプションはスタティックピクチャーとライブラリーピクチャー、さらにピクチャーを使用するアクティブオブジェクトに利用できます (ミラー効果 (Windows)参照)。

このオプションは以下の場合のみ効果があります:

- アプリケーションモード
- Windows
- フォームで"right-to-left"モードが有効になっている

## スタティックテキスト中で参照を使用する

動的なインターフェースを使用するために、テーブルやフィールドのスタティックラベルとして、フィールドや変数の現在値を使用することができます。

以下のタイプのスタティックテキストにダイナミックラベルを使用できます:

- スタティックテキストエリア (テキストやグループボックス)
- フォームウィンドウ名
- ヘルプTipsとヘルプメッセージ (データ入力の制御と補助参照)

**注:** XLIFF (XML) アーキテクチャーに基づく動的参照は、特にインターフェースの翻訳に適していて、これらのラベルやメニュー、ボタンラベル等にも使用できます。詳細は付録 C: XLIFFアーキテクチャーを参照してください。

フォームエディターで**オブジェクト**メニューから**名前表示**または**リソース表示**を選択すると、ラベルの表示を切り替えることができます。

### フィールドや変数の値

スタティックテキストエリアにフィールド名や変数名を埋め込むことができます。テキストエリアが表示または印刷される際、カレントレコードのフィールドや変数の値が使用されます。埋め込みフィールドや変数は、郵便の差し込み印刷、レポートのヘッダーやフッターに使用できます。

フィールド名や変数名は **< >** 記号の間に記述します。

- データベース中のフィールドをいずれも使用できます。カレントテーブルのフィールドの場合、テーブル名を指定する必要はありません。つまり以下のように記述できます: **<FieldName>**。他のテーブルのフィールドを使用する場合、テーブル名を指定する必要があり、以下のように記述します: **<[TableName] FieldName>**。フォームを印刷する際、テキストエリア中の<FieldName>要素はレコードごとにそのフィールドの値で置き換えられます。
- 変数は以下のように記述します: **<VariableName>**。この変数にはオブジェクトメソッドやフォームメソッドで値を割り当てなければなりません。

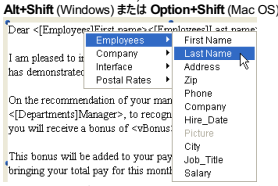
フィールド名や変数名の後にセミコロンと表示フォーマットを記述して、埋め込んだフィールドや変数の値をどのように表示するかを指定できます。例えば埋め込み変数**<vTotal;$###,##0.00>**では、各人の総支払額が計算されてvTotalに代入され、$フォーマットを使用して表示されます。表示フォーマットについては 表示フォーマット を参照してください。

埋め込みフィールドや変数を使用して差し込み印刷をする例は差し込み印刷ドキュメントを作成するを参照してください。

#### フィールド参照メニューを使用する

スタティックテキストブロックにフィールド参照を挿入するために、ポップアップメニューを使用できます:

1. テキストツール を選択する。
2. 以下のキーを押しながらスタティックテキストエリアをクリックする:
   - **Alt** (Windows) または **Option** (Mac OS) を押すと、そのテーブルのフィールドによるポップアップメニューが表示される (これはテーブルフォームのみで利用できます)。
   - **Alt+Shift** (Windows) または **Option+Shift** (Mac OS) を押すと、テーブルおよびフィールドの階層メニューが表示される。




4Dは<FieldName> または <[TableName]FieldName> をテキストエリア中の挿入ポイントに挿入します。

### テーブルとフィールド名

(スタティックテキストとして) ダイナミックなテーブル名とフィールド名をフォームに挿入できます。フォームにダイナミックテーブル名やフィールド名を挿入すると、ストラクチャーエディターでテーブル名やフィールド名を変更したり、4DのSET TABLE TITLESやSET FIELD TITLESコマンドが呼び出された場合、それらの名称がデータベース全体で使用されます。この機能は特にアプリケーションストラクチャーのユーザーに表示される部分の翻訳に適しています。

- ダイナミックテーブル名を挿入するには: **<?[TableName]>** または **<?[2]>** (テーブル番号)
- ダイナミックフィールド名を挿入するには: **<?[TableName]FieldName>** または **<?[2]3>** (テーブルおよびフィールド番号)、または **<?3>** (カレントテーブルのフィールド番号)

**注:** ダイナミックフィールド名はフォームウィザードの詳細画面の**オプション**ページで指定することができます (フォームウィザードを使用してフォームを作成する参照)。

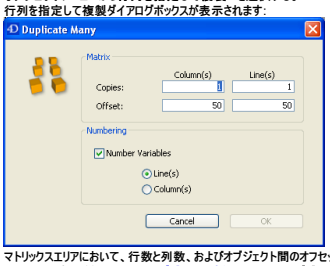
## グリッド上にオブジェクト作成

フォーム上に同じようなアクティブオブジェクトを一度に複数配置し、それぞれの名前がユニークになるように順次番号を指定したい場合があります。例えば、データベース処理を実行する一連のボタンを作成したい場合です。グリッド上にオブジェクト作成を使用すると、複数のオブジェクトを素早く簡単に整列させることができます。

アクティブオブジェクトをマトリックス（格子）上に手動で複製するか、または行列を指定して複製ダイアログボックスを使用することができます。このダイアログボックスにより、素早くマトリックス上にオブジェクトを作成することができます。

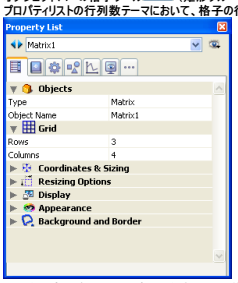
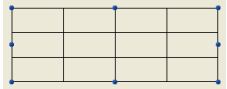
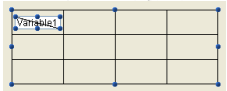
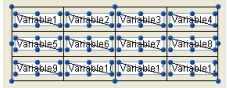
### 行列を指定して複製ダイアログボックスを使用する

行列を指定して複製ダイアログボックスを使用して、1 つ以上のオブジェクトを複製するには:

1. 複製したいオブジェクトを選択する。
2. **オブジェクト**メニューから**行列を指定して複製...**を選択する。
   行列を指定して複製ダイアログボックスが表示されます:
   
   
3. マトリックスエリアにおいて、行数と列数、およびオブジェクト間のオフセットを入力する。
   この点に関する詳細は[フォームオブジェクトの挿入と配置](#)の"オブジェクトを複製する"を参照してください。
4. "変数番号"オプションを選択する。
   このオプションは変数を選択した場合のみ有効になります。
5. 番号振り付け順を選択する。
   行オプションを選択すると、4Dは左から右へ向けて行ごとに、アクティブオブジェクトの番号を設定します。
   列オプションを選択すると、4Dは上から下へ向けて列ごとに、アクティブオブジェクトの番号を設定します。
6. **OK**ボタンをクリックする。
   設定に従い、オブジェクトが複製され、番号が振られます。

### 格子上にオブジェクトを複製する

1. オブジェクトバーの格子ツール （矩形グループの1 つ）を選択し、フォームに格子を作成する。
2. プロパティリストの行列数テーマにおいて、格子の行数と列数を設定する。
   
   
3. 背景色と境界線テーマのオプションを使用して、格子の外観を指定する (任意)。
   境界線スタイル、線幅、カラーと塗りパターンを設定することができます。
4. 格子上のセルの大きさが複製しようとするオブジェクトを格納するために十分であることを確認する。
   
5. 新しいアクティブオブジェクトを作成し、それを格子の左上セルに配置する。
6. オブジェクトと格子の両方を選択する。
   
   
7. **オブジェクト**メニューから**グリッドにオブジェクト作成**を選択する。
   4Dは格子内の各セルにアクティブオブジェクトをコピーし、それぞれのアクティブオブジェクトにユニークな番号を設定します。
   
   

オブジェクトとともにサイズやスタイルのプロパティ、および関連メソッド（該当する場合）もコピーされます。
4D はアクティブオブジェクトに対し、列ごとに上から下へ向けて番号を付けます。各オブジェクトの名前にこの番号が付加され、格子の各セルに名前の重複しないオブジェクトが作成されます。

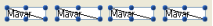
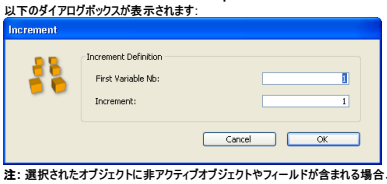
**注:** アクティブオブジェクトに対し、行ごとに左から右へ向けて番号を付けるには、**Alt**キー（Windows）または**Option** キー（Mac OS）を押しながら**グリッドにオブジェクト作成**コマンドを選択します。

メソッド内では、指定された名前でこれらのオブジェクトを参照することができます。この格子は削除することも、フォーム上に残しておくこともできます。

## 変数名を一括で連番にする

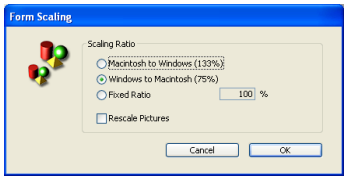
フォームに作成された変数グループ内で、自動に変数名に番号を振る機能が提供されています。このショートカットはたとえばフォームボタンの再構成や、フォーム変数名がユニークであることを保証したい場合に使用できます。この機能はオブジェクトに割り当てられた変数名を変更するものであり、オブジェクト名は変更しないことに留意してください。

変数名をインクリメントするには:

1. 番号を付け直す変数が割り当てられたアクティブオブジェクトを選択する。
   すべてのタイプのアクティブオブジェクトを選択できます。
   
2. Windowsではオブジェクト上で**Ctrl+Alt+クリック**する。
   Mac OSではオブジェクト上で**Command+Option+クリック**する。
   以下のダイアログボックスが表示されます:
   
   
   **注:** 選択されたオブジェクトに非アクティブオブジェクトやフィールドが含まれる場合、このダイアログボックスは表示されません。
3. 開始番号と増分を設定する。
4. **OK**をクリックする。
   設定されたパラメターに基づき、選択されたオブジェクトに番号が振られます。番号は現在の変数名の後ろに追加されます。番号は左から右、上から下に向かって付けられます。

## フォームのスケール

"フォームスケール"ダイアログボックスを使用してフォームオブジェクトの倍率を変更し、データベースが他のプラットフォームで使用されたときの見栄えを良くすることができます。




実際のオブジェクトのサイズが一緒であっても、Mac OSで作成されたフォームオブジェクトはWindowsでは小さく見え、逆の場合は大きく見えます。これはWindowsの画面解像度がMac OSのそれよりも約25%大きいからです。例えばMac OS上の12ポイントのテキストはWindows上で9ポイントで表示されます。フォントサイズをMac OSに合わせてしまうと、Windowsでは小さすぎることになります。逆にWindowsに合わせると、Mac OSでは大きすぎます。

画面解像度の違いを吸収するため、オブジェクトのスケールを変更する必要があります。**フォーム**メニューから**フォームスケール...**を選択して表示されるダイアログを使用して、一回の操作ですべてのフォームオブジェクトのスケールを変更できます。

以下のオプションから選択できます:

- **Macintosh -> Windows (133%)**: Mac OS 上で4D を使用する場合は、これがデフォルトのオプションになります。Mac OS のスクリーン解像度に合わせて作成されたフォームのサイズを変更して、Windows のスクリーン解像度に合わせて作成されたように見せる場合は、このオプションを使用します。これを実現するため、プログラムはフォームオブジェクトのサイズをすべて1/3 だけ大きくします。例えば、9 ポイントのテキストは12 ポイントになります。
- **Windows -> Macintosh (75%)**: Windows 上で4D を使用する場合は、これがデフォルトのオプションになります。Windows のスクリーン解像度に合わせて作成されたフォームのサイズを変更して、Mac OS のスクリーン解像度に合わせて作成されたように見せる場合は、このオプションを使用します。これを実現するため、プログラムはフォームオブジェクトのサイズをすべて1/4 だけ小さくします。例えば、12 ポイントのテキストは9 ポイントになります。
- **カスタムスケール率**: このオプションを使用すると、"%"入力エリアに入力した割合でフォームのサイズを変更することができます。このオプションにより、Mac OS やWindows プラットフォーム上で例外的なスクリーン解像度に出会った場合でも、フォームが適切に表示されるようにフォームのサイズを変更できます。また、このオプションを用いて、全フォームオブジェクトのサイズを使用中のプラットフォーム用に変更することも可能です。例えば、すべてのオブジェクトのサイズを2 倍にしたい場合は、200% と入力します。サイズを半分にしたい場合は、50%と入力します。
- **ピクチャーもスケーリングする**: このオプションは、デフォルトで選択されていません。通常は、ビットマップ形式のピクチャのサイズを拡大、または縮小すると、外観上は良い結果が得られません。このため、このオプションを選択しない限り、プログラムはフォーム上のスタティックピクチャのサイズを変更しません。その代わり、ピクチャは新しい"相対的中心"位置に移動されます。ビットマップを再スケールしても満足な結果が得られることが分かっている場合や、ビットマップ以外のピククチャを使用している場合は、ピクチャのサイズを変更することができます。

## オブジェクトメソッドを使用する

フォーム上のアクティブオブジェクトにメソッドを指定することができます。フォーム上の個々のオブジェクトに指定されたメソッドは*オブジェクトメソッド*と呼ばれます。

最も一般的なオブジェクトメソッドの使用方法をいくつか次に紹介します:

- データ入力制限を実行する。
- タブコントロール、ポップアップメニュー、ドロップダウンリスト、リストボックス、コンボボックス、階層リスト、階層ポップアップメニュー等のインタフェースオブジェクトを初期化し、管理する。
- オブジェクトをクリックまたはダブルクリックすると実行されるアクションを指定する。
- ドラッグ&ドロップ操作を管理する。

データ処理を実行する簡単な例題を次に示します。

次のメソッドは、2 つのフィールドのデータを元にして合計を計算します:

```
Line_Total:=[Products]Price *[Orders]Quantity
```

次のメソッドはNameフィールドのすべての文字を大文字に変換します:

```
[Customers]Name:=Uppercase([Customers]Name)
```

次のメソッドはFirst NameフィールドとLast Nameフィールドの値を連結し、その結果をvNameという名前の変数に代入します:

```
vName:=[Employees]First Name+" "+[Employees]Last Name
```

各オブジェクトメソッドはオブジェクトに割り当てられるため、フォームエディタからオブジェクトメソッドを作成します。メソッドの使用方法については メソッドの編集を参照してください。

### オブジェクトイベント

オブジェクトメソッドは、特定のイベントが発生すると実行されます。例えば、タブコントロールに関連付けたアクションは、ユーザがタブをクリックした場合にのみ意味を持ちます。スクロールエリアでは、ユーザが項目をダブルクリックした場合にのみメソッドを実行したいかもしれません。
プロパティリストの"イベント"テーマにおいて、特定のオブジェクトに対して実行されるイベントを指定することができます。リストには、選択されたオブジェクトに関連するフォームイベントがすべて納められます。一部のイベントは、特定タイプのオブジェクトにのみ利用可能です。また、フォームレベルでしか生成されないイベントもいくつか存在します。フォームイベントに関する説明は4Dランゲージリファレンスマニュアルの Form eventを参照してください。

プロパティリストのイベントテーマにおいて、特定のオブジェクトに対して実行されるイベントを指定することができます。すべてのイベントを選択あるいは非選択したい場合は、**Ctrl** (Windows) または**Command** (Mac OS) キーを押しながらイベントをクリックします。

複数の異なるイベントに対して別々のコードセグメントを実行する必要があれば、メソッド内でCase of...Else...End caseを使用し、イベントエリアで選択した各イベントをテストします。任意のイベントをテストするには、Form event関数とForm Events定数を使用します。オブジェクトメソッドのテンプレートは以下のようになります:

```
C_LONGINT($event)

$event:=Form event

Case of
    :($event=On Load)
  // フォームが開かれたときに実行するコードをここに書く

    :($event=On Data Change)
  // オブジェクトの内容が更新されたときに実行するコードをここに書く

    :($event=On Validate)
  // フォームが受け入れられるときに実行するコードをここに書く

End case
```

### オブジェクトメソッドを作成する

オブジェクトメソッドを作成するには:

1. フォームエディターにおいて、メソッドを割り当てたいオブジェクトを選択する。
2. プロパティリストのアクションテーマ内、オブジェクトメソッド行にある**編集...**ボタンをクリックする。
   または
   **オブジェクト**メニューから**オブジェクトメソッド**を選択する。
   または
   コンテキストメニューから**オブジェクトメソッド**を選択する。
   または
   **Alt** キー(Windows)または**Option** キー(Mac OS)を押しながらフィールドまたはオブジェクトをクリックする。
   4D は新しいメソッドエディターウインドウを表示します(デフォルトで空白)。このメソッドには、"オブジェクトメソッド:"の後ろにフォーム名、そしてオブジェクト名またはフィールド名を付加した名前が付けられます。
3. メソッドの編集で説明する方法でメソッドを作成し、ウィンドウを閉じるか、作業を保存します。
   メソッドがフィールドまたはアクティブオブジェクトに関連付けられます。メソッドが割り当てられたオブジェクトは"オブジェクトメソッド"バッジで確認できます:

   
   デザインモードでメソッドを表示し編集することができます。

### オブジェクトメソッドを開く

オブジェクトメソッドを開くには:

1. フォームエディターで、メソッドを開きたいオブジェクトを選択する。
2. プロパティリストのアクションテーマにあるオブジェクトメソッド行の隣の**編集...**ボタンをクリックする。
   または
   **オブジェクト**メニューから**オブジェクトメソッド**を選択する。
   または
   コンテキストメニューから**オブジェクトメソッド**を選択する。
   または
   **Alt** (Windows) または**Option** (Mac OS) キーを押しながらオブジェクトをクリックする。
   メソッドエディターが開き、メソッドが表示され、編集可能となります。

### オブジェクトメソッドを削除する

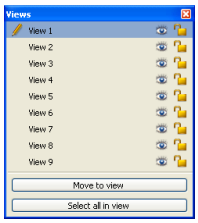
不要になったオブジェクトメソッドはいつでも削除できます。これを行うにはメソッドが割り当てられたオブジェクトを選択し、**オブジェクト**メニューの**オブジェクトメソッド消去**を選択します。

## オブジェクトビューを使用する

4Dフォームでビューを使用できます。必要に応じて表示/非表示が可能な異なるビューにそれぞれオブジェクトを配置することで、複雑なフォーム作成が容易になります。例えばタイプごと (フィールド、変数、スタティックオブジェクト等) に異なるビューに分けることができます。サブフォームやプラグインエリアを含むすべてのタイプのオブジェクトをビューに含めることができます。

### 動作

フォームごとに9つのビューを利用でき、デフォルトでビュー1からビュー9まで名前が付けられています (名前は変更可能)。それぞれのビューごとに表示/非表示とロックが切り換えられます。ビューの管理はビューパレットを使用して行います:




このパレットを表示するには、ウィンドウツールバーのビューボタン をクリックするか、エディターのコンテキストメニューまたはフォームメニューから**ビューリスト**コマンドを選択します。

ビューを使用するためのルールは以下の通りです:

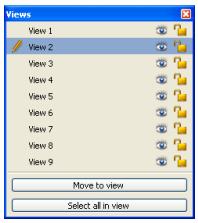
- **利用のコンテキスト**: ビューは純粋に表示上のツールであり、フォームエディター上のみ効果があります。プログラムからビューにアクセスしたり、アプリケーションモードで使用したりすることはできません。
- **ビューとページ**: あるビューには異なるフォームページ上のオブジェクトを含めることができます。ビューの設定にかかわらず、カレントページ (およびページ0) のオブジェクトのみが表示されます。
- **ビューとレベル**: ビューはオブジェクトレベルから独立しています。異なるビューの間で表示上の階層はありません。
- **ビューとグループ**: カレントのビューに属するオブジェクトのみをグループ化できます。

**注:** デフォルトで作成されるフォームやフォームウィザードで作成されたフォームは、タイトルがビュー2に、アクションボタンがビュー3に置かれます。

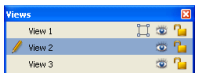
### ビューにオブジェクトを配置する

1つのオブジェクトは1つのビューに配置できます。フォームに作成されたオブジェクトはカレントのビューに属します。デフォルトでビュー1が選択されていて、すべてのオブジェクトはビュー1に属します。

他のビューにオブジェクトを作成するには、事前にビューパレットで目的のビューを選択しておきます:

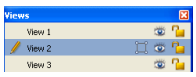



(複数の) オブジェクトを他のビューに移動することもできます。これを行うにはビューを変更したいオブジェクトをフォームエディターで選択します。それらのオブジェクトが属するビューが、ビューリスト上に記号で示されます:




**注:** 選択したオブジェクトは異なるビューに属していることがあります。

次に移動先のビューを選択し、**ビューに移動**ボタンをクリックします。選択したオブジェクトは新しいビューに移動されます:

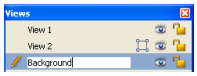



**注:** バッジを使用してオブジェクトがどのビューに属しているかを知ることができます。詳細はバッジを使用するを参照してください。

### ビューの名前を変更する

ビューの名前は変更できます。例えばオブジェクトグループの特徴を表す名前を付けることができます。データベースフォームごと9つのビューにそれぞれ異なる名前を付けることができます。

ビューの名前を変更するには、ビュー名上で**Ctrl+クリック** (Windows) または **Command+クリック** (Mac OS) するか、選択されたビュー名上でダブルクリックします。名前が編集可能になります:




### ビューを使用する

オブジェクトをビューに配置したら、ビューパレットを使用して以下のことが行えます:

- 同じビュー内のすべてのオブジェクトを1クリックで選択できます。
- ビューごとに表示/非表示を切り替えられます。
- ビューのオブジェクトをロックできます。

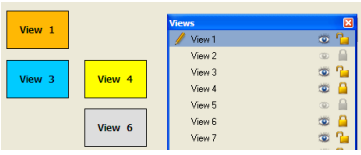
#### ビューのすべてのオブジェクトを選択する

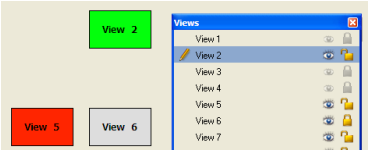
フォームのカレントページ上で、同じビューに属するすべてのオブジェクトを選択できます。この機能は一連のオブジェクトに同じ変更を適用する場合に便利です。
これを行うにはビューを選択し、**ビュー内をすべて選択**をクリックします。

#### ビューのオブジェクトを表示/非表示にする

フォームのカレントページでビューに属するオブジェクトの表示/非表示を切り替えることができます。この方法で、フォームを編集中に特定のオブジェクトにフォーカスすることができます。

デフォルトですべてのビューが表示されていて、それはそれぞれのビューごとに アイコンで示されています。ビューを隠すにはこのアイコンをクリックします。アイコンが薄暗くなり、対応するビューに属するオブジェクトが非表示となります:






**注：** カレントビュー (ビューリスト中で選択されたビュー) は非表示にできません。

非表示のビューを表示にするには、ビューを選択するか、アイコンをクリックします。

#### ビューのオブジェクトをロックする

ビュー中のオブジェクトをロックして選択、変更、削除ができないようにできます。ロックされると、オブジェクトをクリックやマウスドラッグ、**同種のオブジェクトを選択コマンド**で選択することはできません。この機能は操作ミスを防ぐのに役立ちます。

パレットの各ビューの行に表示される アイコンで示されるとおり、デフォルトですべてのビューはロックされていません。ビューのオブジェクトをロックするにはこのアイコンをクリックします。南京錠が閉まり、ビューがロックされます：

**注：** カレントビュー (ビューリストで選択されたビュー) はロックできません。

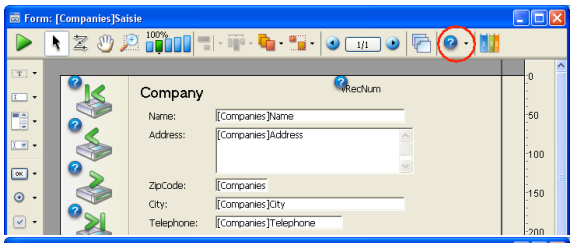
ビューのロックを解除するには、ビューを選択するか、ロックアイコンをクリックします。

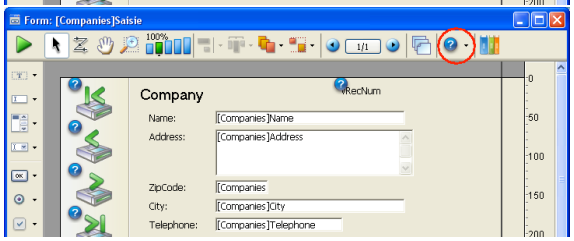
## バッジを使用する

フォームエディターでは**バッジ**を使用してオブジェクトプロパティの表示を容易にできます。

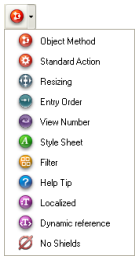
この機能は以下のように動作します。各バッジは特定のプロパティに対応しています (例えばヘルプTipはヘルプが割り当てられていることを示します)。
バッジを有効にすると、4Dは小さなアイコン (バッジ) を、プロパティが割り当てられた各フォームオブジェクトの左上に表示します。

- ヘルプTipのような"true/false"プロパティの場合、オブジェクトに対してそのプロパティが有効である場合にバッジが表示されます。
- ビュー番号のような"値"タイプのプロパティの場合、オブジェクトごとにバッジにその値が表示されます。




バッジを有効にするには、バッジ選択ボタンから表示したいバッジを選択します:




バッジを表示したくない場合、**バッジなし**を選択します。

**注:** アプリケーション環境設定のフォームページで、デフォルトで表示するバッジを設定できます。

各バッジの説明は以下の通りです:

| アイコン | 名前 | 表示 |
|---|---|---|
| | オブジェクトメソッド | オブジェクトメソッドが割り当てられたオブジェクト |
| | 標準アクション | 標準アクションが割り当てられたオブジェクト |
| | リサイズ | 少なくとも1つのリサイズプロパティが割り当てられたオブジェクト、カレントプロパティの組み合わせを表す |
| | 入力順 | 入力可能なオブジェクトで、入力順を表示する |
| | ビュー番号 | すべてのオブジェクトで、ビュー番号を表示する |
| | スタイルシート | スタイルシートが割り当てられたオブジェクト |
| | フィルター | 入力フィルターが割り当てられた入力可オブジェクト |
| | ヘルプTip | Tipが割り当てられたオブジェクト |
| | ローカライズ済み | ラベルに参照が割り当てられたオブジェクト (":"で始まるラベル)、参照はリソース (STR#) またはXLIFFタイプ |
| | ダイナミック参照 | フィールドやテーブル、変数へのダイナミック参照を含むオブジェクト ("<label>"タイプのシンタックス) |
| | バッジなし | バッジは表示されない |

## 設定済みライブラリを使用する

設定済みオブジェクトライブラリは4D フォームにオブジェクトを追加することを簡単にするためにデザインされたツールです。既に設定されたオブジェクトのコレクションが提供され、ドラッグ＆ドロップやコピー/ ペーストするだけで、フォーム中で使用できます。

### ライブラリーオブジェクト

このライブラリーの利用目的は、特定のプロパティを事前に定義することで、標準の4Dオブジェクト (ボタン、テキストエリアなど) の利用を手助けすることにあります。例えば"パスワード入力エリア"オブジェクトは特定のスタイルシートが割り当てられたテキスト変数です。またライブラリーでは日付ピッカーや時刻ピッカーウィジェットなどのハイレベルなオブジェクトも提供されます (4D Widgets（ウィジェット）参照)。

ユーザオブジェクトライブラリ (カスタムオブジェクトライブラリの作成と使用参照) と異なり、設定済みオブジェクトライブラリは変更できません。このライブラリにオブジェクトを足したり、削除したりすることはできません。

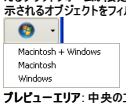
### ライブラリーを使用する

設定済みオブジェクトライブラリーは別ウィンドウに表示されます。オブジェクトはフォーム上にドラッグ＆ドロップで挿入できます。設定済みオブジェクトラ

イブラリーウィンドウを表示するには、4D フォームエディタツールバーの ボタンをクリックします。

設定済みオブジェクトライブラリーが新しいウィンドウに表示されます。この中にはフィルターエリア (ポップアップメニューとボタン)、プレビューエリア、コメントエリアがあります:

- **オブジェクトフィルター**: オブジェクトはいくつかのカテゴリーに分けられています ( ボタン、入力エリアなど)。カテゴリーに属するオブジェクトのみを表示するにはポップアップメニューから選択します。**すべてのカテゴリー**を選択するとすべてのオブジェクトが表示されます。いくつかのオブジェクトは使えるプラットフォームが限定されます (Windows またはMacintosh)。ウィンドウの右上に配置されたボタンを使用してプラットフォームに基づき表示されるオブジェクトをフィルターできます。




- **プレビューエリア**: 中央のエリアにはプレビューおよびオブジェクトの名前が表示されます。オブジェクトをクリックしてオブジェクトに関する情報を得ることができます。説明はウィンドウの下部に表示されます。プレビューエリア上部のスライドを使用して、サムネイルのサイズを変更できます。
- **コメントエリア**: 選択されたオブジェクトに関する情報を表示します。

ウィンドウ中央部からのドラッグ＆ドロップまたはコピー/ペーストを使用してフォームにオブジェクトを挿入できます。オブジェクトは定義済みプロパティとともに配置されます。必要に応じてプロパティを変更することが可能です。

# カスタムオブジェクトライブラリの作成と使用

4Dでオブジェクトライブラリーを作成し、使用することができます。オブジェクトライブラリーは4Dが作成する外部ファイルで、(ボタン、テキスト、ピクチャー、階層リスト等の) すべてのタイプのオブジェクトを格納することができます。そしてこれらのオブジェクトを他のフォームで再利用できます。オブジェクトはプロパティとともに格納されます。ライブラリーはドラッグ＆ドロップやコピー/ペースト操作で利用できます。

ライブラリーを使用すると、グラフィックファミリーや振る舞いごとにグループ化したフォームオブジェクトを作成できます。ライブラリーは外部ファイルに格納されるので、他のデータベースからも簡単に利用できます。

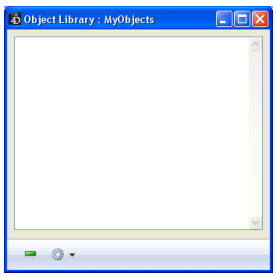
さらにカスタマイズされたデータベースで、ユーザーが編集できるユーザーフォームにオブジェクトライブラリーを提供することができます。この点についてはユーザーフォームで説明しています。

**オブジェクトライブラリーを作成する**

オブジェクトライブラリーを作成するには、**ファイル**メニューまたはツールバーから**新規>オブジェクトライブラリー...**を選択します。

標準のファイルを保存ダイアログボックスが表示され、オブジェクトライブラリーの名前と保存先を指定できます。

ダイアログボックスを受け入れると、4Dはディスク上に新しいオブジェクトライブラリーを作成し、ウィンドウに表示します (デフォルトで空です)。




データベースごとに必要なだけライブラリーを作成できます。Mac OS上で作成されたライブラリーをWindowsで使用すること、あるいはその逆もできます。

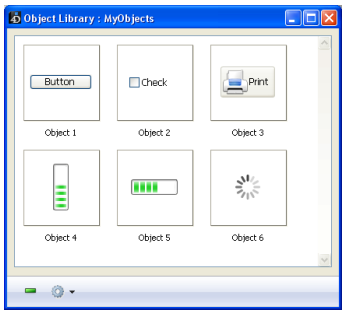
**オブジェクトライブラリーを構築する**

ドラッグ＆ドロップやコピー/ペースト操作で、オブジェクトをオブジェクトライブラリーに配置できます。フォームあるいは他のオブジェクトライブラリーからコピーできます (設定済みライブラリを使用するを含む)。元のオブジェクトとのリンクは保持されません。オリジナルが編集されてもライブラリー内のオブジェクトには影響しません。

**注:** フォームからオブジェクトライブラリーにドラッグ＆ドロップ操作を行うためには、4D環境設定で"ドラッグ＆ドロップを開始"を選択しなければなりません (フォームページ参照)。

追加、削除、名称変更などの基本操作はコンテキストメニューあるいはウィンドウのオプションメニューから行えます:

オブジェクトライブラリーには個々のオブジェクトやオブジェクトグループを格納できます。それぞれのオブジェクトは1つのアイテムとしてグループ化されます:

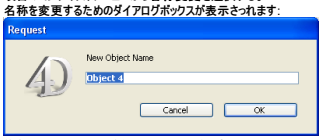



1つのオブジェクトライブラリーには32,000項目まで含めることができます。

オブジェクトはグラフィックおよび動作両方すべてのプロパティとともにコピーされます。これらのプロパティはオブジェクトがフォームや他のライブラリーにコピーされる際にも保持されます。

**オブジェクトの名称変更**

新規項目には"Object"にランダムな番号を追加した名前が付けられます (例えば*Object2012*)。この名前は必要に応じて変更できます。名前を変更するには:

- 項目をダブルクリックする。
- 項目のコンテキストメニューから**名称変更**を選択する。
  名称を変更するためのダイアログボックスが表示されます:
  
  
  複数の項目に同じ名前を付けることもできます。

**依存オブジェクト**

コピー/ペーストやドラッグ＆ドロップで特定のライブラリーオブジェクトを使用すると、依存オブジェクトもコピーされます。例えばボタンをコピーするとそのオブジェクトに割り当てられていたオブジェクトメソッドもコピーされます。これらの依存オブジェクトはそれのみを直接コピーしたりドラッグ＆ドロップしたりすることはできません。

メインのオブジェクトが使用されるのと同時にライブラリーにペーストされる依存オブジェクトは以下の通りです:

- リスト
- スタイルシート
- フォーマット/フィルター
- ピクチャー
- ヘルプTips (フィールドにリンク)
- STR#リソース
- オブジェクトメソッド

オブジェクトに使用されている文字列だけでなく、STR#リソース全体がコピーされます。

**オブジェクトライブラリーを開く**

既存のオブジェクトライブラリーを開くには、**ファイル**メニューまたはツールバーから**開く>オブジェクトライブラリー...**コマンドを選択します。

標準のファイルを開くダイアログボックスが表示さっれ、開くオブジェクトライブラリーファイルを選択できます。オブジェクトライブラリーを同時に複数のデータベースで開くことはできません。他方1つのデータベースで複数のオブジェクトライブラリーを開くことはできます。ライブラリーの拡張子は".4il"です。

**オブジェクトライブラリーにアクセスする**

オブジェクトライブラリーには4Dデザインモードおよび (デベロッパーが許可していれば) アプリケーションモードのユーザーフォームエディター (ユーザーフォーム参照) からアクセスできます。すべてのライブラリー項目がデザインモードで表示可能です。しかしユーザーフォームエディターではオブジェクトメソッドが割り当てられていない項目のみが表示されます。

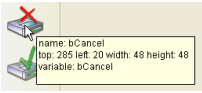
ひとつでもオブジェクトメソッドが割り当てられたオブジェクトがある項目にはバッジ (オブジェクトメソッド月のオブジェクトに表示されるものと同じ、バッジを使用する参照) が表示され、名前は青になります。これによりデザインモードでデベロッパーに表示されるが、アプリケーションモードでユーザーに表示されないものを見分けることができます。

#### リモートモードでの動作

すべてのクライアントからオブジェクトライブラリーにアクセスできるようにするには、ライブラリーファイルをデータベースのResourcesフォルダーに置かなければなりません (4Dファイルの説明参照)。このフォルダーはサーバーとクライアントマシン間で (ピクチャーやXLIFFなど) カスタム要素を共有するために使用されます。詳細は4D Serverリファレンスマニュアルのリソースフォルダの管理を参照してください。
Resourcesフォルダーの内容はリソースエクスプローラーで確認、管理できます。

## フォーム上のオブジェクトに関する情報を実行時に表示する

フォームを実行中に、ショートカットを使用してオブジェクトに関する有用な情報 (名前や座標など) を取得できます。情報は**Ctrl+Shift** (Windows) または **Control+Shift** (Mac OS) を押しながらオブジェクト上にマウスカーソルを合わせると、ヘルプTipに表示されます:




この情報はデザインモードが開かれている時に、フォームに表示されているオブジェクトgと途に利用できます。

# アクティブオブジェクトを使用する


アクティブオブジェクトとは

フィールドおよび変数オブジェクト

ボタン

3Dボタン、3Dチェックボックス、3Dラジオボタン

ピクチャーボタン

ボタングリッド

チェックボックス

ラジオボタンとピクチャーラジオボタン

ポップアップメニュー、ドロップダウンリスト

コンボボックス

階層ポップアップメニューと階層リスト

ピクチャーポップアップメニュー

インジケーター

タブコントロール

スプリッター

Webエリア

プラグインエリア

## アクティブオブジェクトとは

アクティブオブジェクトとは、データベースタスクやインタフェース機能を実行するものすべてを指し、さまざまな種類があります。フィールドはアクティブオブジェクトとして扱われます。他のアクティブオブジェクトとしては入力可オブジェクト（変数）、コンボボックス、ドロップダウンリスト、ピクチャーボタン等があり、これらのオブジェクトはデータを一時的にメモリへ保存したり、またはダイアログボックスを開く、レポート印刷、バックグラウンドプロセスの開始などの動作を実行します。

場合によっては、プロパティリストウインドウでアクティブオブジェクトを選択し、その動作を指定することができます。例えば、内蔵の自動アクションを使用し、ボタンの動作を指定することができます。また別の場合では、オブジェクトと自動的に関連付けられるメソッドを作成して、そのオブジェクトの動作を指定することも可能です。

4Dでは以下のアクティブオブジェクトを使用できます:

- フィールドおよび変数オブジェクト,
- ボタン,
- 3Dボタン、3Dチェックボックス、3Dラジオボタン,
- ピクチャーボタン,
- ボタングリッド,
- チェックボックス,
- ラジオボタンとピクチャーラジオボタン,
- ポップアップメニュー、ドロップダウンリスト,
- コンボボックス,
- 階層ポップアップメニューと階層リスト,
- ピクチャーポップアップメニュー,
- インジケーター,
- タブコントロール,
- スプリッター,
- Webエリア,
- プラグインエリア,
- リストボックス (別の章で説明),
- サブフォームとウィジェット (別の章で説明).

プロパティを使用してアクティブオブジェクトの外観や動作を指定できます。それぞれのオブジェクトタイプごとに使用できるプロパティを説明します。データ入力制御や表示フォーマットなど基本的な汎用プロパティはアクティブオブジェクトのプロパティで説明します。

## フィールドおよび変数オブジェクト

フォーム上でフィールドと変数はほぼ同様に動作します。

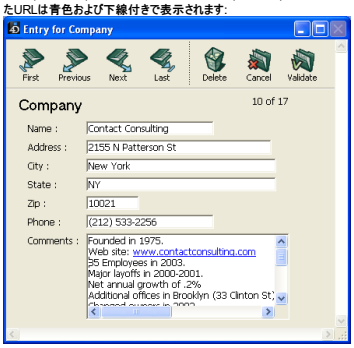
- フィールドはレコードのデータを表示し、また入力するために使用します。フォームウィザードで新規フォームを作成するとき、標準オブジェクトとしてフォームに含めるフィールドを選択します。フォームを作成したら、フォームエディターを使用して、表示フォーマットやデータ入力制御など追加のプロパティを指定できます。これらのプロパティはそれが指定されたフォーム上でのみ有効です。同じプロパティを他のフォームでも指定できますし、別なプロパティを指定することもできます。プロパティを変更したり、追加/削除したりできます。
- 変数は入力可あるいは入力不可の形態をとることができ、テキスト、数値、日付、時間、ピクチャー型等のデータを受け取れます。変数ツール を使用してフォーム上に作成できます。
  プロパティリスト内の「オブジェクト」テーマ内の**変数タイプ**によって変数のデータの型を指定することが出来ます。
  フィールドと同様、変数もデータの型を入力・表示することができます。変数はデータの一時的な保存のために使用されます。例えばよくある使われ方としては、メソッドを使用した計算結果を表示する、といったものです:
  ```
  vTotal:=Quantity*Price
  ```
  計算結果を表示する変数を作成し、その名前をvTotalとしています。そしてメソッドを使用して計算を行います。
  メソッドを使用して入力可あるいは入力不可の変数を管理できます。入力可の変数はデータを受け入れます。フィールドに行うのと同様にデータ入力制御を設定できます。入力されたデータはオブジェクトに割り当てられた変数に代入されます。オブジェクトの変数名を使用してデータを管理できます。

フィールドオブジェクトを変数オブジェクトに、あるいはその逆に変更できます:

- データベースのフィールドからセレクション型の**リストボックス**へと、データを直接表示・入力することができます(リストボックスにフィールドを表示するを参照して下さい)。
- 4D v14以降、ポップアップメニュー、ドロップダウンリスト と コンボボックス オブジェクトを使用することによってリストフィールドまたは変数をフォーム内にて直接表示することが出来るようになりました。

### 表示

- 変数やフィールドのサイズは必要に応じて設定できます。文字を表示する場合、その高さは設定されたフォントのサイズに基づきステップごとに設定できます。変数やフィールドには表示フォーマットを適用できます (表示フォーマット参照)。テキストやピクチャーオブジェクトには縦横のスクロールバーを付けることができ (スクロールバー参照) 可変フレームで印刷することができます (テキストタイプオブジェクトの印刷参照)。
- テキストあるいは文字タイプの変数やフィールドに**マルチスタイル**プロパティが設定されている時、オブジェクトレベルで設定されたスタイルに加え、文字ごとにスタイルを設定することができます。詳細はマルチスタイル (リッチテキストエリア)を参照してください。
- (http, ftp, www または mailtoで始まる) URLはテキストや文字型のフィールドや変数に入力されると自動で認識されます。ユーザーが**Ctrl+クリック** (Windows) または **Command+クリック** (Mac OS) すると、URLはデフォルトのブラウザーで直接実行されます。Windowsでは検知されたURLは青色および下線付きで表示されます:




**注:** 電子メールアドレスはmailto:addressの形式で書く必要があります。

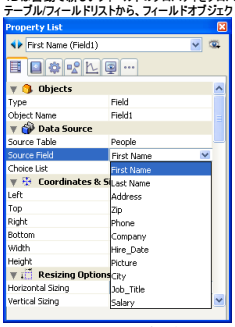
### フォームにフィールドを追加する

いつでもフォーム上にフィールドを追加あるいは削除できます。例えば以下のようなとき、フォームにフィールドを追加する必要が生じます:

- フォームウィザードで選択しなかったフィールドが必要になったとき。
- データベースストラクチャーにフィールドを追加して、それをフォームで使用する必要がある時。

フォームにフィールドを追加するには:

1. オブジェクトツールバーからフィールド挿入ツール を選択し、フォーム上にフィールドを描画する。
   4Dは自動で新しいフィールドのプロパティをプロパティリスト上に表示します。
2. テーブル/フィールドリストから、フィールドオブジェクトに割り当てるフィールドを選択する:




   **注:** BLOB型のフィールドは選択できません。
3. 必要に応じてフィールドにプロパティを指定する。
   フィールドを作成した後、通常追加のプロパティを設定する必要があります。データ入力制御、ヘルプテキスト、メソッド、リサイズ、フォント、アピアランスオプション等を設定できます (アクティブオブジェクトのプロパティ参照)。
   **注:** エクスプローラーのテーブルページからドラッグ＆ドロップでフィールドを挿入することもできます。

新しいフィールドはあなたが配置した場所に表示されます。フィールドエリアには選択したテーブル名とフィールド名が表示されます。4Dはデフォルトでフィールドを示すラベルを配置しません。開発者がスタティックテキストエリアを使用して作成します。ラベルの内容が動的になるよう記述できる点に留意してください (スタティックテキスト中で参照を使用する参照)。

フォーム上にフィールドを配置したら、他のフォームオブジェクトと同様にサイズやフォント、フォントカラー等のプロパティを変更できます。

#### フィールドと変数を相互に変更する

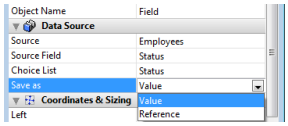
アクティブ、非アクティブ含め、すべてのオブジェクトタイプを他のオブジェクトタイプに変更できます。フィールドは変数に変換でき、変数はフィールドに変換できます。例えばフィールドをフォームに配置した後で、値をレコードとして格納する必要がなくなった場合、そのエリアを変数に変更できます。オブジェクトを他のオブジェクトに変更すると、オブジェクトの元のプロパティ (座標、オブジェクトメソッド、アピアランス、カラーなど) は保持されます。フィールドのデータタイプは変数のデータタイププロパティでも保持されます。ピクチャーフィールドが割り当てられたフィールドエリアを変数に変換すると、プロパティリストの変数タイプはピクチャーになります。

変数をフィールドに、フィールドを変数に変換するには、オブジェクトを選択し、プロパティリストのオブジェクトテーマからタイプドロップダウンリストで**フィールド**または**変数**を選択します。プロパティリストは、新しいオブジェクトタイプのプロパティを表示するために更新されます。オブジェクト名、オブジェクトメソッド、またそのプロパティ (サイズ、入力可等) には変更ありません。

変数をフィールドに変更すると、4Dはデフォルトで先頭テーブルの先頭フィールドをオブジェクトに割り当てます。データソーステーマでソーステーブルとソースフィールドを設定できます。

### 関連付け

フィールドまたは変数に選択リストを関連付けた場合、プロパティリストの「データソース」内の関連付けのオプションを使用することが出来ます。




このオプションにより、保存するデータのサイズを最適化できるようになります。詳細な情報に関しては、関連付け(値または参照番号)を参照して下さ

[illegible]
い。

**フォーム上のピクチャー変数の型宣言**

フォーム上のピクチャー変数にピクチャーを表示する際には特別なメカニズムが使用されます。そのために変数を設定する際には、その変数がピクチャーを表示するために使用されることを明確に指定しなければなりません。この指定はフォームがロードされる前 (On Loadフォームイベントよりも前に) に行う必要があります。

これを行うには、以下の方法があります:

- フォームを呼び出す前 (例えばDIALOGコマンドを呼び出す前) にC_PICTURE(varName) を実行する
  または
- プロパティリストの変数タイプポップアップメニューからピクチャーを選択する

これのいずれかを行わない場合、ピクチャー変数は正しくピクチャーを表示できません (インタープリターモードのみ)。

# ボタン

## 概要

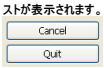
フォームエディターを使用し、さまざまなボタンをフォームに追加することができます。フォームにボタンを追加すると、各ボタンに標準アクションを割り当てることができます。自動動作ボタンにより、レコードの入力、取り消し、削除、レコード間の移動、マルチページフォームでのページ間の移動、サブフォームレコードのオープン、削除、追加等の操作を行うことができます。

通常は、フォームウィザードを使用してフォームを作成する際にボタンを追加します。これらのボタンの動作はプロパティリストで変更することができます。例えば、ボタンから標準アクションを取り除いたり、またはボタンの動作を指定するオブジェクトメソッドを作成することができます。

またフォームエディターを使用してボタンを追加したり、ボタンのアクションを指定することも可能です。例えばフォーム上で複数のサブフォームが必要になる場合は、必要なサブフォームと自動ボタンをフォームエディターで追加できます。各ボタンをフォームに追加し、それぞれのボタンに標準アクションを関連付けます。

4Dでは次のようなボタンを使用することができます:

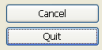
- **ボタン**: これらのボタンは現在のプラットフォームインタフェースで表示されます。選択したフォント、フォントサイズ、書体、カラーを用いてボタンテキストが表示されます。

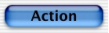
  ボタンに表示されるラベルは、プロパティリストの"オブジェク"トテーマにある**タイトル**フィールドで設定します。ボタンラベルはいつでも変更することができます:

- **デフォルトボタン**: 境界線が少し太いことを除けば、デフォルトボタンは標準のボタンとまったく同じにみえます。これはユーザーに対しこのボタンが推奨できる選択であることを示します。次の図は、デフォルトボタンと標準のボタンとを比較しています。

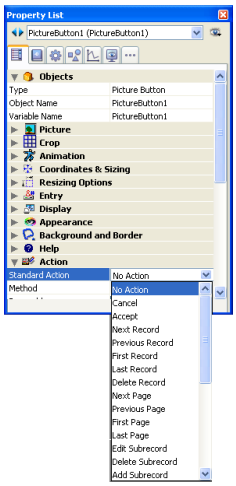
  Mac OSの場合、デフォルトボタンは点滅します:

  **デフォルトボタン**というオブジェクトタイプは存在しませんが、ボタンで使用できるプロパティです。
  **注:** フォームページごとに、デフォルトボタンは1つしか作成できません。
- **ハイライトボタンと透明ボタン**: これらのボタンは、グラフィックオブジェクトの上部に配置するために設計されています。ハイライトボタンは透明です。ハイライトボタンをクリックすると、そのボタンのグラフィックが反転表示されます。
  一方、透明ボタンは常に透明であり、クリックされても反転表示されません。別ページの表示など、クリックにより実行される動作により、そのボタンがクリックされたことがわかります。透明ボタンは、ボタン機能を表わすテキストやグラフィックの上部に配置すべきです。ユーザがテキストやグラフィックをクリックすると、そのボタンがアクティブになります。
- **3Dボタンとピクチャーボタン**: 3Dボタンファミリー(3Dボタン、3Dチェックボックス、3Dラジオボタン)とピクチャボタンには、さまざまな固有のプロパティがあります。これらのボタンについては3Dボタン、3Dチェックボックス、3Dラジオボタンとピクチャーボタンを参照してください。

## ボタンの有効化を管理する

フォーム実行時、標準アクションが設定されたボタンは必要に応じてグレー表示されます。例えばあるテーブルの1番目のレコードが表示されると、先頭レコードボタンと前レコードがグレー表示されます。
タイプドロップダウンリストから目的のボタンタイプを選択して、ボタンを作成します。この後標準アクションドロップダウンリストから標準ボタンの動作を選択することができます (標準アクション参照):




**注:** エディターのコンテキストメニューから**標準アクション**階層メニューコマンドを使用することも可能です。

ボタンのタイプが何であれ、このドロップダウンリストから項目を選択しなければなりません。標準アクションドロップダウンリストに表示されていない動作をボタンに実行させたい場合は**動作なし**を選択して、ボタンの動作を指定するオブジェクトメソッドを記述します (メソッドの編集参照)。

通常はイベントテーマでOn Clickedイベントを有効にして、ボタンのクリック時にのみメソッドを実行します。どのタイプのボタンにもメソッドを割り当てることができます。

ボタン(標準ボタン、ハイライトボタン、透明ボタン、ラジオボタン、ラジオピクチャ、チェックボックスなど)に関連付けられたすべての変数は、デザインモードやアプリケーションモードでフォームが初めて開かれるとき0に初期化されます。ボタンをクリックすると、変数は1 になります。

それぞれのアクションについては標準アクションを参照してください。

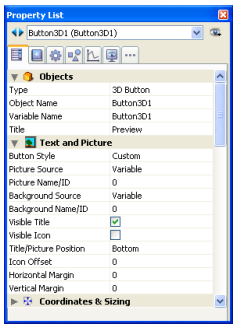
**互換性に関する注記:** 4Dの以前のバージョンでは、標準アクション付きボタンがコンテキストによりグレー表示に切り替わる際の副作用を使用して、カスタムボタンの有効/無効を自動で切り替えることができました (標準アクション付きボタン変数と同じ名前の変数名をカスタムボタンにも使用した場合)。この副作用は4D v12で取り除かれました。各ボタンの有効/無効は個別に処理しなければなりません。

## 3Dボタン、3Dチェックボックス、3Dラジオボタン

3Dボタンファミリーには3Dボタン、3Dチェックボックス、3Dラジオボタンがあります。構造上これらのオブジェクトは同じですが、関連付けられた変数の処理方法だけが異なります:

- フォームが開かれた時、3Dボタンに関連付けられた変数の値は0です(標準状態)。ユーザがボタンをクリックするとこの値は1になります(押された状態)。この後標準状態へ戻り、値は再び0になります。
- 3Dチェックボックスに関連付けられた変数の値は、ボックスが選択されていない場合は0で、選択されると1になります。ボタンとは異なり、3Dチェックボックスは再び選択されるまでその状態(0 または1)が変わりません。詳細はチェックボックスを参照してください。
- 3Dラジオボタンはグループ単位で動作します。選択されたボタンに関連付けられた変数の値は1 になり、それ以外は0 になります。フォームを開いた時、変数の値は0または1のいずれかになります。詳細はラジオボタンとピクチャーラジオボタンを参照してください。

3Dボタンファミリー特定のプロパティが数多く用意され、これらのプロパティを使用して、各種オペレーティングシステムへ完全に統合できるインタフェースを設定することができます。具体的には3Dボタン(ベベルボタン、プッシュボタン等)に対して、さまざまな定義済みスタイルを適用したり、ポップアップメニューを割り当てることができます。各種プロパティを組み合わせて指定すれば、数々のバリエーションが実現します。




**タイトル**

このプロパティを使用すると、ボタンにラベルを追加することができます。このラベルのフォントとスタイルは"テキスト"テーマで指定できます。
\文字 (バックスラッシュまたは円マーク) を使用すると、ラベル内で強制的に改行することができます。

ラベルに \ を表示したい場合は \\ と入力します。

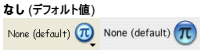
デフォルトでラベルはボタンの中央に置かれます。またボタンにアイコンも含まれている場合、**タイトル/ピクチャー位置**プロパティを用いて、これら2 つの要素の相対位置を変更することができます。

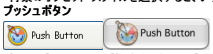
**タイトル表示**プロパティの選択を解除すると、タイトルを隠すことができます。この場合、アイコンは自動的にボタンの中央へ移されます。

インターフェースの翻訳の目的で、XLIFFやSTR#参照をボタンのタイトルエリアに入力することができます (付録 C: XLIFFアーキテクチャー参照)。

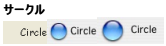
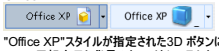
**ボタンスタイル**

このプロパティを使用するとボタンの全般的な外観を設定することができます。またスタイルによっては、特定のオプションが利用できなくなることもあります。次のスタイルを使用することができます:

- **なし** (デフォルト値)
  
  
  "なし"スタイルが指定された3Dボタンは、透明ボタンに似ています。つまり、ボタンの"反転表示"が行われません。しかし、3D ボタンのオプションは利用することができます。
  Mac OSの場合、ポップアップメニューがリンクされていることを示す三角形を表示することはできません。
- **背景のオフセット**

  このスタイルはハイライトボタンに対応しています。ただしボタン背面のピクチャーをオフセットすることにより、ユーザーがこのタイプのボタンをクリックした場合に3D効果が得られます。
  "背景のオフセット"スタイルを選択すると、いずれのオプションも使用できません。
- **プッシュボタン**
  
  
  プッシュボタンスタイルが指定された3D ボタンは、標準のシステムボタンとして表示されます。また、このスタイルでは"ポップアップメニューあり"プロパティ以外の3D ボタンオプションを利用することもできます。
- **ツールバーボタン**

  
  このスタイルの3Dボタンは、主としてツールバーで使用するためのものです。
  Windows の場合、ボタンはマウスのロールオーバー時に反転表示されます。"ポップアップメニューあり"プロパティを使用すると、ボタンの右側中央に三角形が表示されます。
  Mac OS の場合ボタンは反転表示されません。"ポップアップメニューあり"プロパティを使用すると、ボタンの右下部分に三角形が表示されます。
- **カスタム**

  
  この3D ボタンスタイルではカスタム背景ピクチャーを使用でき、さまざまな追加パラメターを管理することができます(アイコンやマージンのオフセット)。詳細は後述の"カスタム3Dボタン"を参照してください。
  このスタイルでは"ポップアップメニューあり"プロパティ以外の3D ボタンオプションを利用することができます。
- **サークル**
  
  
  Mac OS X の場合、"サークル"スタイルが指定された3Dボタンは円形のシステムボタンとして表示されます。サークルには2 種類の規定スタイルが利用できますが、これはフォーム上のボタンのサイズによって異なります。
  このスタイルには"ポップアップメニューあり"プロパティ以外の3D ボタンオプションを利用することができます。
  Windows の場合、このボタンスタイルは"なし"スタイルと同じです(背景にあるサークルは考慮されません)。
- **スモールシステムスクエア**
  Mac OS の場合、"スモールシステムスクエア"スタイルが指定された3D ボタンはシステムスクエアとして表示されます。このスタイルでは"ポップアップメニューあり"プロパティ以外の3D ボタンオプションを利用することができます。
  Windows の場合、このボタンスタイルは"なし"スタイルと同じです(背景にあるスクエアは考慮されません)。
- **Office XP**
  
  
  "Office XP"スタイルが指定された3D ボタンには次のような特徴があります:
  - 反転表示と背景のカラーはシステムカラーに基づいています。
  - Windows で使用する場合、マウスのロールオーバー時にのみ反転表示が行われます。

  このスタイルでは3Dボタンオプションを利用することができます。
- **ベベル**

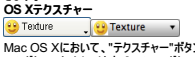
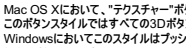
  
  Mac OS の場合、"ベベル"ボタンは標準のシステムボタンとして表示されます。このボタンでは"ポップアップメニューあり"プロパティを含めて、3D ボタンオプションを利用することができます。
  Windows の場合、このボタンスタイルは"ツールバーボタン"スタイルと似ていますが、ポップアップメニューが関連付けられていることを示す三角形がボタンの右下に表示される点だけが異なります。
- **角の丸いベベル**

  
  Mac OS の場合、"角の丸いベベル"ボタンは"ベベル"ボタンと似ていますが、反転表示されると丸みを帯びた形になる点が異なります。
  Windows の場合、このボタンスタイルは"ベベル"スタイルと同じです。
- **折りたたみ/展開**

  このボタンは標準の折りたたみ/展開アイコンを追加するために使用できます。これらのボタンは階層リストで使用されます。Windowsではボタンは[+] や[-]で表示されます。Mac OSでは右や下を指す三角として表示されます。このスタイルは3Dチェックボックスでのみ使用されます (ボタンの

状態がチェックボックスの選択/選択解除に対応します)。

- **ヘルプ**

  このボタンスタイルはシステム標準のヘルプボタンを表示するために使用します。このスタイルを使用するとフォームにシステムヘルプボタンを追加できます。
- **OS Xテクスチャー**
  
  
  Mac OS Xにおいて、"テクスチャー"ボタンは灰色のグラデーションを表示する標準のシステムボタンです。高さは定義済みで、変更できません。このボタンスタイルではすべての3Dボタンオプションを使用できます。
  Windowsにおいてこのスタイルはプッシュボタンと同じですが、ポップアップメニューを持つことができ、Vistaにおいては透明になることができる特別機能を持っています。
- **OS Xグラデーション**
  
  
  Mac OS Xにおいて、"グラデーション"ボタンは2トーンのシステムボタンです。このボタンスタイルではすべての3Dボタンオプションを使用できます。
  Windowsにおいてこのスタイルはプッシュボタンと同じですが、ポップアップメニューを持つことができます。

### 3D ボタンにアイコンを追加する

3D ボタンスタイルにアイコンを付加することができます("背景のオフセット"スタイルを除く)。

関連付けたアイコンの管理は、複数のプロパティ(ピクチャーソース、ピクチャ名/ID)を用いて行います。

- **ピクチャーソース**
  ピクチャーボタンと同様に、3D ボタンのアイコンは異なる4種類のソース(**変数**、**ピクチャーライブラリー**、**リソースファイル**、**ファイル**)から参照します。ソースを設定したら、"ピクチャー名/ID"プロパティにピクチャーの名前または番号を指定することができます。
- **ピクチャー名/ID**
  ピクチャーソースを設定したら、このエリアにピクチャーの名前(ピクチャーが変数である場合やピクチャーライブラリから参照する場合)、番号(ピクチャーライブラリーまたはリソースファイルからピクチャーを参照する場合)、またはパス名 (ピクチャーをピクチャーファイルから参照する場合)を入力します。ピクチャーファイルの場合、パス名はデータベースのResourcesフォルダーからの相対パスで記述しなければなりません (ピクチャーファイルの自動参照参照)。3Dボタンピクチャーを追加したくない場合、このプロパティに0を入力します。
  **注:** ピクチャーライブラリーやディスクファイルからのドラッグ&ドロップでボタンにピクチャーを割り当てることもできます。
- **状態の数**
  このプロパティは3Dボタンのアイコンとして使用されるピクチャーに含まれる状態の数を指定します。一般的にボタンアイコンは4つの状態、アクティブ、クリック、ロールオーバー、無効を含んでいます。ソースピクチャーでは、状態を表すアイコンは縦に並んでいなければなりません:

- **タイトルを表示 / アイコンを表示**
  ボタンにタイトルとピクチャー両方が含まれるとき、タイトルを表示やアイコンを表示の選択を外すことで、それらを非表示にできます。いずれかが隠されると、他方が中央に表示されます。

### タイトル/ピクチャー位置

このプロパティを使用すると、関連するアイコンに対するボタンタイトルの相対的な位置を変更することができます。ボタン内に、タイトル(関連ピクチャーなし)またはピクチャー(タイトルなし)しか含まれていない場合、このプロパティは無効になります。3D ボタンにタイトルとピクチャーが含まれる場合、デフォルトとしてテキストはピクチャーの下部に置かれます。

このプロパティの各種オプションの結果を次に示します:

- **左**: テキストはアイコンの左側に置かれます。ボタンの内容は右揃えされます。

  
- **上**: テキストはアイコンの上側に置かれます。ボタンの内容は中央揃えされます。

  
- **右**: テキストはアイコンの右側に置かれます。ボタンの内容は左揃えされます。

  
- **下**: テキストはアイコンの下側に置かれます。ボタンの内容は中央揃えされます。

  
- **中央**: アイコンのテキストはボタンの縦と横に中央揃えされます。例えば、テキストをアイコンの中に組み込む場合は、このパラメータが役立ちます。

### ポップアップメニューあり

このプロパティを使用すると、3D ボタン内で三角形として表われるシンボルを表示することができます。このシンボルは、ポップアップメニューが付属することを示します:

このシンボルの外観と位置は、ボタンのスタイルと現在のプラットフォームによって変わります。"ポップアップメニューあり"プロパティを利用できる3D ボタンのスタイルは次の通りです:

- なし
- ツールバーボタン
- ベベル
- 角の丸いベベル
- Office XP
- OSXテクスチャー
- OSXグラデーション

#### リンクと分離

ポップアップメニューシンボルを3D ボタンに付加する際に、**リンクと分離**という2つのオプションから選択することができます。

- 分離
- リンク

**注:** 実際に"分離"モードを利用できるかどうかは、ボタンのスタイルとプラットフォームによって決まります。

それぞれのオプションにより、ボタンとポップアップメニューとの関係が指定されます:

- ポップアップメニューが分離している場合、ボタンの左側部分をクリックするとボタンのカレントアクションが直接実行されます。このアクションは、ボタンの右側からアクセスできるポップアップメニューを使用して変更することができます。
- ポップアップメニューがリンクしている場合、ボタンをクリックしてもポップアップメニューが表示されるだけです。このポップアップメニュー上のアクションを選択しないと、実行は行われません。

またこれらのオプションは、ボタンのフォームイベント管理にも影響を与えます (詳細は4Dランゲージリファレンスマニュアルを参照)。

#### ポップアップメニューの管理

"ポップアップメニューあり"プロパティは、ボタンのグラフィック面だけを管理するという点に注意してください。ポップアップメニューとその値の表示は、すべて開発者が処理しなくてはなりません。具体的にはフォームイベントやDynamic pop up menu、Pop up menuコマンドを使用してこれを処理します。

### カスタム3D ボタン

"カスタム"スタイルの3D ボタンを選択すると、さらに複数のプロパティ(バックグランドソース、バックグランド名/ID、アイコンオフセット、水平マージン、垂直マージン)が使用できるようになります。

- **バックグランドソース**: このプロパティを使用すると、ボタンの背景に描かれるピクチャーを設定することができます。アイコンと同様に、変数、ピクチャーライブラリー、リソースファイル、ディスクファイルのいずれからピクチャーを参照するかを指定できます。
- **バックグラウンド名/ID**: ソースの設定後、このエリアにピクチャーの名前や番号を指定することができます。
  アイコンと同様に、バックグラウンドピクチャーには4 つの個別の縦のエリアを含むことができます。これはボタンの標準的な4つの状態(アクティブ、クリック、ロールオーバー、無効)を表わすために使われます。クリックされた場合の動作は**アイコンオフセット**プロパティを使用して処理することもできるという点に注意してください。
- **アイコンオフセット**: このプロパティを使用すると、独自のオフセット値をピクセル単位で設定することができます。この値はボタンのクリック時に使用され、入力したピクセル数だけボタンのタイトルが右下へシフトされます。この機能により、ボタンのクリック時に独自の3D 効果を適用することができます。
- **水平マージン / 垂直マージン**: これらのプロパティを使用すると、ボタンの内側のマージンサイズ(ピクセル単位)を設定することができます。これらのマージンにより領域が区切られ、3D ボタンアイコンとタイトルはこの領域を超えて配置されなくなります。
  例えば、背景ピクチャーに境界線が含まれる場合は、このパラメターが役立ちます:

マージンなしのカスタム3D ボタン

13 ピクセルのマージンが指定された3D ボタン

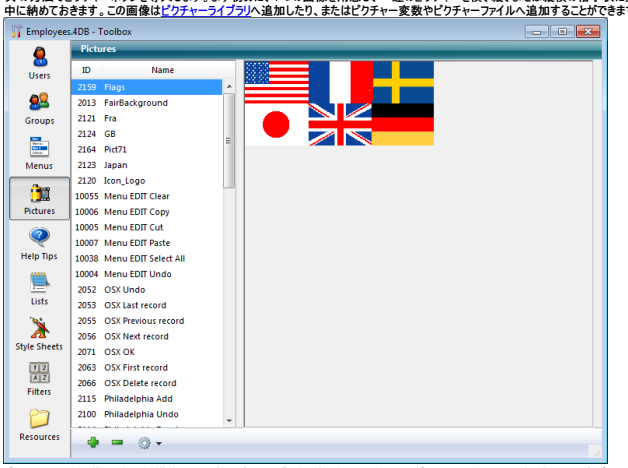
## ピクチャーボタン

### 使用方法

ピクチャーボタンは複数の状態を持てます。比較してみると、標準ボタンは3 つの状態（有効、無効、クリック）に対応します。ピクチャーボタンでは、その名前が表わすようにそれぞれの状態を別々のピクチャーにより表わします。

ピクチャーボタンは、次の2つの方法で使用します:

- フォーム上のコマンドボタンとして。この場合ピクチャーボタンには通常4種類の状態があります（有効、無効、クリック、ロールオーバー）。これはフォームウィザードのほとんどのフォームテンプレートで用いられる設定です。
- 複 数の選択項目の中からユーザーに選ばせるためのピクチャーボタンとして。この場合、ピクチャーボタンをポップアップピクチャーメニューの代わりに使用する ことができます。ピクチャーポップアップメニューではすべての選択肢が (ポップアップメニューの項目として) 同時に表示されます。一方ピクチャーボタンは、選択候補を連続的に表示します（ボタンをクリックする度に変わります）。
  次に示すのは、ピクチャーボタンの例です。例えばカスタムアプリケーションのユーザに、アプリケーションのインタフェース言語を選ばせたいものとします。そこで下図のように、選択候補をピクチャーボタンとしてカスタムプロパティダイアログボックスに組み込みます:

  
  オブジェクトをクリックするとピクチャーが変わる
  次 の方法でピクチャーボタンを導入します。まず初めに、1 つの画像を用意し、一連のピクチャーを横、縦、または縦横の格子状に並べてその中に納めておきます。この画像はピクチャーライブラリへ追加したり、またはピクチャー変数やピクチャーファイルへ追加することができます。
  
  
  ピクチャーは、縦、横、または縦横格子状に整理することができます（前の図を参照）。ピクチャーを格子状に並べた場合、各ピクチャーには左から右へ1 行ずつ、0から始まる番号が振られます。例えば、4 つの行と3 つの列で構成される格子において、2 行目の2 番目の画像の番号は4になります。
  **注**: ピクチャーライブラリには、ピクチャーをサムネールテーブルとして整理できる機能があります。ピクチャーがサムネールテーブルとして定義されていれば、ライブラリーからピクチャー名をドラッグすることでピクチャーボタンを作成することができます。

### プロパティ

ピクチャーボタンには次のような固有のプロパティがあります:

- "ピクチャー"テーマにおいて、"ソース"ドロップダウンリストを使用してピクチャーソースを指定します。変数、ピクチャーライブラリー、リソースファイル、ファイルのいずれかを選択することができます。ピクチャーソースを設定したら、ピクチャーの名前（ピクチャーが変数の場合）、番号（ピクチャーがピクチャーライブラリーリソースファイルから参照される場合）、またはパス名 (ピクチャーファイルのピクチャーを参照する場合）を「名前/ID」エリアに入力します。ファイルの場合、パス名はデータベースの**Resources**フォルダーからの相対パスで記述しなければなりません (ピクチャーファイルの自動参照参照)。
- "行列数"テーマには、サムネールテーブルを構成する行数と列数を指定することができます。本文の例では3列2行からなるピクチャーを使用しています。
- "アニメーション"テーマには、ピクチャーボタンの表示モードと動作を指定することができます。いくつかのオプションを組み合わせることが可能です。この件については次の節で詳しく説明します。
  **注**: ピクチャーボタンに関連付けた変数は、ピクチャーのサムネールテーブルで現在表示されているピクチャーのインデックス番号を返します。このテーブル内のピクチャー番号は0 から始まります。

### アニメーション

ピクチャーボタンに指定できる表示モードと動作モードを次に示します。通常は、これらのオプションを組み合わせて使用します:

- **<オプション未選択>**
  ユーザがクリックすると、系列中の次のピクチャーを表示します。**Shift** キーを押しながらクリックすると、系列中の前のピクチャーを表示します。系列中の最後のピクチャーに達すると、もう一度クリックしてもピクチャーは変わりません。つまり、この設定では、系列中の最初のピクチャーへ一巡して戻ることはありません。
- **マウスプレス中は自動更新**
  最初のオプションと似ていますが、ユーザがマウスボタンを押して各ピクチャーを連続的に表示できる点が異なります（つまり、アニメーションのように表示する）。最後のピクチャーに達しても、オブジェクトは最初のピクチャーに戻りません。
- **先頭フレームへ戻る**
  最初のオプションと似ていますが、ピクチャーが連続的に繰り返して表示される点が異なります。最後のピクチャーに達して再度クリックすると、最初のピクチャーが表示され、そのまま続きます。
- **ロールオーバー効果**
  マウスカーソルが通過すると、ピクチャーボタンの内容が変わります。カーソルがボタンエリアを離れると、最初のピクチャーが再度表示されます。このモードは、マルチメディアアプリケーションやHTML ドキュメントで頻繁に利用されます。DISABLE 時は最終フレームオプション（128）を選択していなければ、この際に表示されるピクチャーはサムネールテーブル中の最後のピクチャーです。このオプションを選択した場合は、最後から二番目のサムネールが表示されます。
- **マウスアップで戻る**
  このモードは2 つのピクチャーを用いて動作します。ユーザがボタンをクリックした時以外は一番目のピクチャーが常に表示されます。ボタンがクリックされると、マウスボタンが放されるまで二番目のピクチャーが表示され、マウスボタンが放されるとすぐに一番目のボタンへ戻ります。このモードを使用すると、それぞれの状態（アイドルとクリック）ごとに異なるピクチャーを使用した動作ボタンを作成することができます。このモードを使い、3D 効果を作成したり、ボタンの動作を表現するピクチャーを表示することができます。
- **DISABLE時は最終フレーム**
  このモードを使用すると、ボタンが無効な場合に表示するサムネールとして、最後のサムネールを設定することができます。このモードを選択すると、4Dはボタンが使用不可になると、最後のサムネールを表示します。ボタンが使用不可の場合に用いられるサムネールは、4Dにより別に処理されます。このオプションと"マウスプレス中は自動更新"オプションおよび"先頭フレームへ戻る"オプションを組み合わせると、最終ピクチャーはボタンに割り当てられた順序から外され、無効時にのみ表示されるようになります。
- **アニメーション間隔（tick）**
  このモードを使用すると、一定のスピードで（tick 単位）ピクチャーボタンの内容が繰り返し表示されます。例えば10 を渡した場合、サムネールは10 tick ごとに変わります。このモードでは、他のすべてのオプションが無視されます。

**注**: 透過オプション（"アピアランス"テーマ）を使用して、ピクチャーボタンの描画方法を設定することも可能です（ピクチャーの背景を透明にする）。

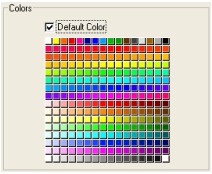
例えばマウスアップで戻る、ロールオーバー効果、DISABLE 時は最終フレームモードを受け入れるボタンを設定したいものとします。
1 行4 列からなるサムネールテーブルの場合、各サムネールはデフォルト、クリック、ロールオーバー、無効という状態に対応しています。
プロパティリストにおいて次のプロパティを設定します。1 行4 列、マウスアップで戻る、ロールオーバー効」、DISABLE 時は最終フレームオプション。

## ボタングリッド

### 使用方法

ボタングリッドは透明なオブジェクトであり、画像の最前面に配置されます。この画像は列と行の配列を表現するべきです。ボタングリッドのオブジェクトを使用すると、ユーザがグラフィック上でクリックした場所を判別することができます。オブジェクトメソッドではOn Clickedイベントを使用し、クリックされた場所に応じて適切な動作を実行します。

4Dでは、カラーパレットにボタングリッドが使用されています:




グリッド上のボタンには、左上から右下に向けて番号が振られます。この例でグリッドは16列×16行で構成されています。左上にあるボタンはクリックされると1 を返します。2 行目の右端にある赤いボタンが選択されると、ボタングリッドは32 を返します。

ボタングリッドを作成するには、背景グラフィックをフォームに追加し、そのグラフィックの最前面にボタングリッドを配置します。行列数テーマの該当する入力エリアに**行数**と**列数**を指定します。

### ページ指定アクション

ボタングリッドに**ページ指定**アクションを割り当てることができます。この動作を選択すると、4D はボタングリッドで選択されたボタンの番号に相当するフォームページを自動的に表示します。
例えばグリッド上の10 番目のボタンを選択すると、4D は現在のフォームの10 ページ目を表示します（存在する場合）。
任意のボタンの選択結果を自分で管理したい場合は、**動作なし**を選択します。

詳細は標準アクションを参照してください。

# チェックボックス

## 使用方法

チェックボックスを使用して、バイナリデータ(true-false)の入力や表示を行います。これはボタンタイプの一つです。チェックボックスの状態は選択、未選択のいずれかになります。その効果はメソッドで管理します。他のボタンと同じように、フォームが初めて開かれた時は、チェックボックスが0 に初期化されています。チェックボックスが選択されると、チェックボックスに割り当てられたメソッドが実行されます。

チェックボックスは小さな四角形の右側にテキストを表示します。このテキストはプロパティリストのオブジェクトテーマにあるタイトルエリアで設定します。このエリアにXLIFFまたはSTR#参照の形式でラベルを指定できます (MissingRef参照)。

ユーザがオブジェクトをクリックすると、ボックスが選択されます。チェックボックスが選択されると、その値は1 になります。チェックボックスが選択されていない場合、その値は0 になります。フォーム上のすべてのチェックボックスは選択/未選択のいずれかの状態になります。

チェックボックスのグループを使用すると、複数の候補を選択できるようになります。

チェックボックスとしてフォーマットされたブールフィールドとは異なり、チェックボックス変数の値は自動的に保存されません。メソッドを用いてこの変数を管理します。

**注:** 3D チェックボックスの動作はチェックボックスと同じですが、その外観(例えば選択済み/未選択状態の表現方法)は3D ボタンファミリーのプロパティで設定されます。詳細は3Dボタン、3Dチェックボックス、3Dラジオボタンを参照してください。

## 3 ステートチェックボックス

チェックボックスオブジェクトは3 番目の状態を受け入れます。この3 番目の状態とは中間的な状態であり、一般的には表示のために用いられます。この状態を使用すると、例えばあるプロパティは全オブジェクトの中ではなく、選択オブジェクト群の中に存在する、ということを表現することができます。

チェックボックスでこの3 番目の状態を制御するには、プロパティリストの"表示"テーマにおいて**3 ステート**プロパティを設定しなくてはなりません。:

このプロパティは、数値タイプの変数に関連付けられた標準のチェックボックスに対してのみ使用できます。3D チェックボックスやブールフィールドのチェックボックスには3 ステートプロパティを利用することができません(ブールフィールドは中間状態になりません)。

チェックボックスが3 番目の状態になると、チェックボックスに関連付けられた変数は値2を返します。

**Tip:** 入力モードにおいて3 ステートチェックボックスはチェックなし / チェック / 中間状態 / チェックなしという順序でそれぞれの状態を順に表示します。一般的にこの中間状態は入力モードではあまり役に立たないため、2 という値になった場合は、コード上で変数の値を強制的に0 に設定し、チェックされた状態からチェックなしの状態へ直接移行してください。

## ラジオボタンとピクチャーラジオボタン

ラジオボタンとピクチャーラジオボタンは、ボタングループやピクチャーグループの中から1つを選択することができるオブジェクトです。ラジオボタンは、小さな目玉状の円とテキストを表示します。ピクチャーラジオボタンはアイコンまたはピクチャーを表示します。これらはグラフィックの最前面に配置されます。

**注：** 3D ラジオボタンの動作はラジオボタンと同じですが、その外観は3D ボタンファミリーのプロパティで設定されます。詳細は3Dボタン、3Dチェックボックス、3Dラジオボタンを参照してください。

各タイプのラジオボタンは同じ方法で選択することができます。つまり、オブジェクトをクリックして選択します。また、選択されたピクチャーラジオボタンをクリックすると選択が解除されますが、ラジオボタンの場合はそれが行えません。

ピクチャーラジオボタンは、選択されるまでボタンが透明であるという点でハイライトボタンと似ています。ボタンが選択されると、別のラジオボタンが選択されるまで、背面にあるピクチャーを反転表示します。

この節ではこれ以降"ラジオボタン"という文言を使用して、すべてのタイプのラジオボタンを表わします。

ラジオボタンは組織的な集合の中で使用されます。その集合のなかから一度にひとつのボタンしか選択できません。一連のラジオボタンが連携した状態で動作するためには、フォームエディター上でそれらがグループ化されていなくてはなりません。グループ化を行うには**オブジェクト**メニューの**グループ化**コマンド、またはフォームのツールバーの該当するボタンを使用します。

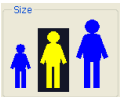
以前のバージョンの4D では、ボタンに関連付けた変数の名前に同じ頭文字を指定することにより、一連のラジオボタンの動作を連携させることができました（例えば、m_button1、m_button2、m_button3 など）。互換性の理由から、変換後のデータベースではこの方式がデフォルトとして維持されています。ただしデータベース設定 (互換性ページ参照) で、強制的に新しいモードを使用するよう指定することができます。

ラジオボタンの結果はメソッドを用いて管理します。あらゆるボタンと同様に、フォームが初めて開かれる時ラジオボタンは0またはFalse (割り当てられた変数の型による) に初期化されています。ラジオボタンが選択されると、ラジオボタンに割り当てられたメソッドが実行されます。

次の例は、ビデオ収集データベースで3D ラジオボタンを使用し、レコーディングの速さ（SP、LP、EP）を入力しているところです。

グループのなかから1つのラジオボタンを選択すると、そのボタンには1またはTrueが代入され、グループ内の他のすべてのボタンには0またはFalseが代入されます。一度に1つのラジオボタンしか選択できません。

次の例は、ピクチャーラジオボタンです。選択されたピクチャーラジオボタンは黒背景で表示されます:




ラジオボタンオブジェクトに格納された値は (ブールフィールドである場合を除き) 自動保存されません。変数に格納されたラジオボタンの値はメソッドで管理しなければなりません。

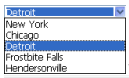
# ポップアップメニュー、ドロップダウンリスト

## 概要

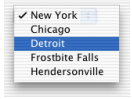
ポップアップメニューやドロップダウンリストは、リストから選択を行えるようにするためのオブジェクトです。ポップアップメニューに表示される項目は配列を用いて管理します。

"ポップアップメニュー"および"ドロップダウンリスト"という名前は、同じタイプのオブジェクトのことを意味します。"ポップアップメニュー"はMac OS の用語であり、"ドロップダウンリスト"はWindows の用語です。

次の例に示すように、これらのオブジェクトの外観はプラットフォームによって若干異なります:



Windows



Mac OS

## 配列の使用

配列とは、メモリー内の値の一覧のことで、配列の名前によって参照されます(配列とフォームオブジェクト を参照して下さい)。ポップアップメニュー・プルダウンリストは、クリックするとその配列を値の一覧として表示します。

このオブジェクトを初期化するには、値のリストを配列にロードします。複数の方法で初期化を行うことができます:

- プロパティリストでデフォルト値のリストを入力します。これを行うには、プロパティリストのデータソーステーマにおいて"デフォルト値"フィールドの**編集...**ボタンをクリックします。詳細についてはデフォルト値のリストを参照してください。このデフォルト値は、配列へ自動的にロードされます。オブジェクトに関連付けた変数名を使用して、この配列を参照することができます。
- オブジェクトが表示される前に、値を配列要素に代入するコードを実行します。例えば:

```
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"
```

  この場合フォームのオブジェクトに関連付けた変数名は*aCities*でなければなりません。
  このコードはフォームメソッド内に置かれ、On Loadフォームイベントが発生すると実行されます。
- オブジェクトが表示される前にLIST TO ARRAYコマンドを使用してリストの値を配列にロードします。例えば:

```
LIST TO ARRAY("Cities";aCities)
```

  この場合フォームのオブジェクトに関連付けた変数名は*aCities*でなければなりません。
  このコードは、前述した代入命令文の代わりに実行できます。

ユーザが行った選択内容をフィールドに保存する必要があれば、レコードの登録後に代入命令を実行します。これを行うには、オブジェクトメソッドに次のようなCase文を作成します:

```
Case of
   :(Form event=On Load)
      LIST TO ARRAY("Cities";aCities)
      If(Record number([People])<0) //新レコード
         aCities:=3 //デフォルトの値を表示
      Else //既存のレコード、保存された値を表示
         aCities:=Find in array(aCities;City)
      End if
   :(Form event=On Clicked) //ユーザーが修正したセレクション
      City:=aCities{aCities} //フィールドの新しい値を代入
   :(Form event=On Validate)
      City:=aCities{aCities}
   :(Form event=On Unload)
      CLEAR VARIABLE(aCities)
End case
```

プロパティリストのイベントテーマにおいて、作成したCase 文の中でテストする各イベントを選択します。

配列には常に有限数の項目が納められます。項目リストは動的であり、メソッドを用いて変更可能です。配列の項目は変更したり、並び替えや追加を行うことができます。

配列の作成と使用に関する詳細は4Dランゲージリファレンスの配列を参照してください。

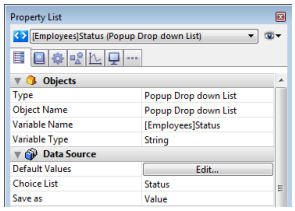
## 選択リストの使用

ポップアップメニュー/プルダウンリストを、列挙型のフィールドまたは変数の管理のために使用したい場合、フィールドまたは変数をオブジェクトのデータソースとして直接参照することができます。これにより列挙型のフィールドまたは変数を容易に管理できるようになります。

**注:** 階層リストの場合、第一階層の値のみが表示・選択できます。

例えば、"White"、"Blue"、"Green"、"Red" という値のみを含む "Color" というフィールドがあった場合、これらの値を含むリストを作成し、それを "Color"フィールドを参照するポップアップメニューに関連付けることができます。こうすることによって、あとは自動的に4Dがカレントの値の入力や表示に関して管理してくれます。

ポップアップメニュー/ドロップダウンリストやコンボボックスをフィールドや変数と関連付けるには、オブジェクトの**変数名** のエリアにフィールドまたは変数の名前を直接入力するだけです:




フォームを実行すると、4D が自動的に入力と表示中のポップアップメニュー/コンボボックスの状態を管理してくれます。ユーザーが値を選択すると、その値はフィールドに保存され、このフィールドの値はフォームが表示されたときにポップアップメニューの中に表示されます:

**注:** この原理は、配列を用いてオブジェクトを初期化する方法と組み合わせることはできません。**変数名** のエリアにフィールド名を入力した場合は必ず選択リストを使用して下さい。

### 関連付け

ポップアップメニュー/ドロップダウンリストを選択リストとフィールドに関連付けると、プロパティリスト内の「データソース」テーマ内にて**関連付け**のオプションを選択できるようになります。

このオプションにより、保存するデータのサイズを最適化できるようになります。詳細な情報に関しては、関連付け(値または参照番号)を参照して下さい。

## ページ指定アクション

**ページ指定** 標準アクションをポップアップメニュー/ドロップダウンリストに割り当てることができます(プロパティリストの"アクション"テーマ)。このアクションを選択すると、4D はドロップダウンリストで選択されたオブジェクトの番号と一致するフォームのページを自動的に表示します。

例えばユーザーがリストの3 番目の項目を選択すると、4Dはカレントフォームの3 ページ目(存在する場合)を表示します。項目を選択した結果をプログラムで管理したい場合は**動作なし**を選択します。

# コンボボックス

コンボボックスはドロップダウンリストと似ていますが、キーボードから入力されたテキストを受けいれる点と、二つの特定のオプションがついている点が異なります。コンボボックスの初期化方法はドロップダウンリストとまったく同じです (ポップアップメニュー、ドロップダウンリスト参照)。

ユーザーがコンボボックスにテキストを入力すると、その値は配列の0番目の要素に代入されます。その他の点では、コンボボックスを入力可エリアと同じように取り扱い、配列または選択リストを一連のデフォルト値として使用します。

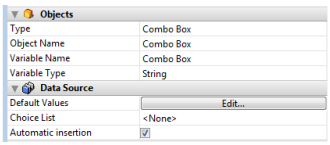
On Data Changeイベントを使用し、入力可エリアのオブジェクトと同じ方法で、入力可能エリアへの入力内容を管理します。詳細については4Dランゲージリファレンスマニュアルの Form eventコマンドの説明を参照してください。

**コンボボックスのオプション**

コンボボックス型のオブジェクトには、関連付けられた選択リストに関係する二つのオプションがあります。**自動挿入**と**除外リスト**(除外された値のリスト)です。

**自動挿入**

**自動挿入**のオプションは、コンボボックス型オブジェクトのプロパティリストの「データソース」テーマ内にあります:




**注:** このオプションはリストボックスの列にも存在します。リストボックスの列のセルは、列が選択リストと関連付けられているときはコンボボックスとして表示されているからです。

このオプションがチェックされていると、オブジェクトに関連付けられた選択リストにない値をユーザーが入力した場合に、その値が自動的にメモリー内のリストに追加されます。 OBJECT SET LIST BY NAME または OBJECT SET LIST BY REFERENCE コマンドを使用して選択リストを関連付けることができます。

例えば、"France, Germany, Italy" という値を含む選択リストが "Countries" というコンボボックスに関連付けられていた場合を考えます。自動挿入のオプションがチェックをされていて、ユーザーが "Spain" という値を入力すると、"Spain" という値が自動的にメモリー内のリストに追加されます:

当然のことながら、オブジェクトに除外リストが設定されていた場合にはそのリストに含まれる値は入力することができません(詳細は以下の章を参照して下さい)。

**注:** デザインモードで定義されたリストが関連付けられた場合、自動挿入によって、オリジナルの選択リストが変更されることはありません。

**自動挿入**のオプションがチェックされていない場合、入力された値はオブジェクトの中には保存されますが、メモリー内のリストには入力されません。

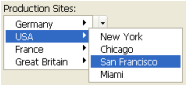
**除外リスト**

**除外リスト**のオプションはコンボボックス型のオブジェクトの"値の範囲"のテーマ内にあります。これを通して除外された値のリストをこれらのオブジェクトに関連付けることができます。ユーザーがこのリストに含まれる値を入力したとき、その入力は自動的に却下されます(除外リストを参照して下さい)。

**注:** 指定リストはコンボボックスには割り当てることはできません。画面上において、オブジェクト内にいくつかの指定された値を表示したいときには、ポップアップメニュー型のオブジェクトを使用して下さい。

## 階層ポップアップメニューと階層リスト

階層ポップアップメニューにはサブメニューがあり、そのメニュー内にはさらに項目が割り当てられています。階層メニューの例を次に示します:




同様、階層リストにはサブリストがあり、そのリスト内にはさらに項目が割り当てられています。階層リストの例を次に示します:

**注:** フォームにおいて階層ポップアップメニューは2レベルに制限されています。しかし階層リストタイプのオブジェクトにはこのような制限はありません。

三角形のアイコンをクリックすると、階層リストを展開したり、折り畳むことができます。

ユーザが階層リストの項目を変更できるかどうかを管理することができます。階層リストの項目が修正可能である場合、**Alt+クリック**(Windows)または**Optionキー+クリック**(Mac OS)ショートットを使用するか、または項目のテキスト上でロングクリックを行うと、編集できるようになります。リストエディターで作成したリストを用いて階層リストを生成する場合は、リストエディターの**編集可能項目**オプションを使用して、階層リストの項目の修正が可能かどうかを管理することができます。詳細はリストプロパティの設定を参照してください。

**注:** リストエディターからドラッグ&ドロップで直接階層ポップアップメニューや階層リストを作成することができます (リストエディターからのドラッグ&ドロップ参照)。

4D ランゲージの階層リストコマンドを使用して、階層ポップアップメニューや階層リストを管理することができます。これを行うには、フォームエディターでオブ ジェクトに関連付けた変数に階層リスト参照を割り当てます。詳細には4Dランゲージリファレンスマニュアルの階層リストを参照してください。

また、OBJECT SET LIST BY REFERENCE コマンドや OBJECT SET LIST BY NAME コマンドを使用することによって、階層リスト参照にフォームオブジェクトの選択リスト(ソース、指定リストと除外リスト)を関連付けることもできます。

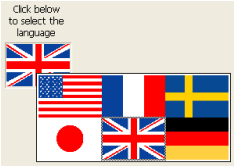
# ピクチャーポップアップメニュー

## 使用方法

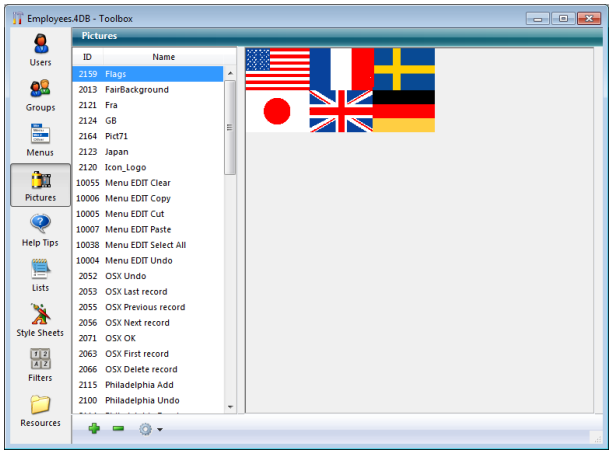
ピクチャーポップアップメニューは、画像の二次元配列を表示するポップアップメニューです。ピクチャーポップアップメニューを使用して、ピクチャー ボタンを置き換えることができます。ピクチャーポップアップメニューで使用するピクチャーの作成方法は、ピクチャーボタン用のピクチャーと似ています。

その概念はボタングリッドと同じですが、グラフィックがフォームオブジェクトではなくポップアップメニューとして使用される点が異なります。

ピクチャーメニューを作成するには、画像を参照する必要があります。次の例は、ピクチャーボタンのために定義されたピクチャーを使用しています (ピクチャーボタン参照)。この場合、ピクチャーポップアップメニューからピクチャーを選択することにより、インタフェース言語を選ぶことができます。各言語は対応する国旗で表わされます:




ピクチャーボタンと同様に、ピクチャーポップアップメニューは、列と行で構成されたピクチャーを使用します。このピクチャーはピクチャーライブラリ、ピクチャー変数、ピクチャーファイルに配置することができます:




そしてピクチャーポップアップメニューをフォームに配置できます。

**注:** ピクチャーライブラリにはサムネールテーブルとしてピクチャーを整理できる機能があります。また現在の設定による結果をプレビューすることも可能です。ピクチャーをサムネールテーブルとして定義すると、**Shift** キーを押しながらピクチャー名をフォームへドラッグして、ピクチャーポップアップメニュー作成することができます。

## プロパティ

さまざまな固有のプロパティを使用して、ピクチャーポップアップメニューの設定を行うことができます:

- "ピクチャー"テーマの"ソース"ドロップダウンリストを使用して、ピクチャーソースを設定します。ピクチャーソースは変数、ピクチャーライブラリー、リソースファイル、ファイルのなかから選択することができます。ピクチャーソースを設定したら、"名前/ID"エリアにピクチャーの名前 (ピクチャーが変数である場合)、番号 (ピクチャーライブラリまたはリソースファイルからピクチャーを参照する場合)、またはパス名 (ファイルのピクチャーを参照する場合) を入力します。ファイルの場合、パス名はデータベースの**Resources**フォルダーからの相対パスで記述しなければなりません (ピクチャーファイルの自動参照参照)。
- "行列数"テーマには、サムネールテーブルを構成する行数と列数を指定します。
- **水平マージン**と**垂直マージン**オプションでメニューの端からピクチャーまでの間のマージンを指定します。値はピクセル単位で指定します。

## ページ指定アクション

ピクチャーポップアップメニューに**ページ指定**アクションを割り当てることができます。このアクションを選択すると、4Dはピクチャー配列で選択されたピクチャー位置に相当するフォームのページを、自動的に表示します。要素は左から右、上から下に向かって番号が割り当てられます。

例えばユーザーが3番目の要素をクリックすると、4Dはカレントフォームの3ページ目 (存在する場合) を表示します。
クリックをプログラムから管理したい場合は"動作なし"を選択します。

標準アクションについては標準アクションを参照してください。

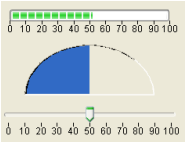
## プログラミング

メソッドを使用してピクチャーポップアップメニューを管理できます。ボタングリッドと同様、ピクチャーポップアップメニューに割り当てられた変数に、選択された要素の値が代入されます。項目が選択されなければ、この値は0 になります。各項目には、一番上の行の左から下へ行ごとに番号が振られます。

# インジケーター

進捗インジケーター (または"サーモメーター")、ルーラー、ダイアルは図形を用いて値を表示するオブジェクトです。これら3 つのオブジェクトはその外観が異なるだけであり、それぞれ同じように機能します。これらのオブジェクトを4Dではインジケーターと呼びます。

インジケーターを使用して値の表示や設定を行うことができます。例えばメソッドを用いてサーモメーターに値を指定すると、その値が表示されます。ユーザがインジケーターポイントをドラッグすると、その値が変更されます。この値はフィールド、入力可オブジェクト、入力不可オブジェクト等の他のオブジェクトで使用することができます。



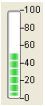
進捗インジケーター / ダイアル / ルーラー

**インジケータのタイプ**

4Dは3つのメインタイプのインジケーターとその派生タイプを提供します。どのタイプのインジケーターを使用するかは"インジケータータイプを決定する"を参照してください。

**進捗バー**

*主タイプ*: 進捗インジケーター (サーモメーター)




進捗バーはデフォルトの進捗インジケーターです。縦または横の進捗バーを表示できます。どちらになるかはフォームエディター上に描かれたオブジェクトの形により決定されます。

**バーバーショップ**

*主タイプ*: 進捗インジケーター (サーモメーター)

(MacOS)

(Windows)

このタイプの進捗インジケーターは連続したアニメーションを表示します。"バーバーショップ"サーモメーターは通常プログラムが何らかの処理を行っていて、それが終了する時間が予測できない場合、そのことをユーザーに通知するために使用します。このバリエーションが選択されると"スケール"テーマは非表示になります。

フォームが実行されたとき、オブジェクトのアニメーションは開始されません。割り当てられた変数に値を代入してアニメーションを管理します:

- 非0値 = アニメーション開始
- 0 = アニメーション停止

**注:** "バーバーショップ"サーモメーターはアピアランスがシステムまたは印刷に設定されている場合にのみ動作します。

**非同期進捗バー**

*主タイプ*: 進捗インジケーター (サーモメーター)

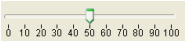
この円形のインジケーターも連続したアニメーションを表示します。このタイプのオブジェクトはネットワーク接続を確立したり、計算を行う作業を実行中である時に使用します。このバリエーションが選択されると"スケール"テーマは非表示となります。

フォームが実行されたとき、アニメーションは開始されません。割り当てられた変数に値を代入してアニメーションを管理します:

- 非0値 = アニメーション開始
- 0 = アニメーション停止

**ルーラー**

*主タイプ*: ルーラー




ルーラーはカーソルを使用して目盛り上を移動することで値を選択する標準インターフェースオブジェクトです。割り当てられた変数の値をフィールドに代入して格納したり、逆にオブジェクトの現在値を設定することもできます。

**数値ステッパー**

*主タイプ*: ルーラー

vStep変数に割り当てられたステッパー

この標準オブジェクトを使用すると、ユーザーは矢印ボタンをクリックすることで、数値、時間、または日付を定義済みのステップ毎にスクロールすることができます。

オブジェクトに割り当てた変数の値をフィールド等に代入して値を格納したり、逆にオブジェクトの現在値を変更したりできます。

ステッパーは直接数値、時間、日付変数に割り当てることができます。

- 時間タイプの値では最小、最大、ステッププロパティは秒を表します。例えばステッパーを8:00から18:00まで10分のステップで設定するには以下のように設定します:
  - 最小 = 28,800 (8*60*60)
  - 最大 = 64,800 (18*60*60)
  - ステップ = 600 (10*60)
- 日付タイプの値ではステッププロパティに入力された値が日数を表します。最小と最大プロパティは無視されます。

**注:** ステッパーを時間や日付変数に対して動作させるためには、プロパティリストで型を設定することとおよびC_TIMEまたはC_DATEコマンドで明示的に宣言する必要があります。

**ダイアル**

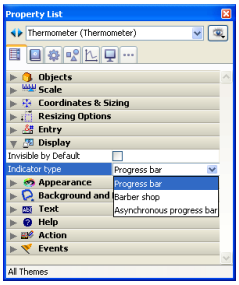
*主タイプ*: ダイアル

このタイプのインジケーターは半円形の中にデータを表示します。

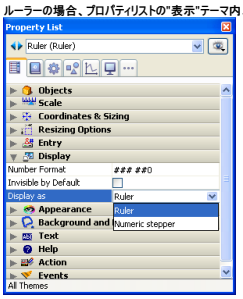
**インジケータータイプを決定する**

ダイアル、進捗インジケーター、ルーラーはフォームエディターのオブジェクトバーの ボタンから利用できます。さらに"進捗インジケーター" (サーモメーター) と "ルーラー" オブジェクトには数種のバリエーションがあります。

- 進捗インジケーターの場合、プロパティリストの"表示"テーマ内、**インジケータータイプ**プロパティを使用してバリエーションを設定します:




- **ルーラーの場合、プロパティリストの"表示"テーマ内、表示タイプ**プロパティを使用してバリエーションを設定します:



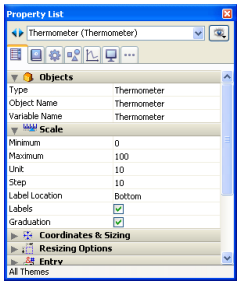

**注:** これらのバリエーションはOBJECT SET FORMATコマンドを使用して設定することもできます。

**インジケータープロパティ**

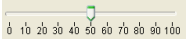
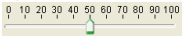
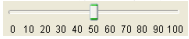
標準の位置調整とアピアランスの設定に加えて、インジケーター固有のプロパティを他にもいくつか設定することができます。つまり最小値、最大値、目盛りの単位、インジケーター最小単位、および表示オプション等のプロパティです。
またインジケーターのラベルの表示フォーマットを定義することも可能です（表示フォーマットに関する詳細はGET LIST ITEM PARAMETERを参照）。

**目盛り**

いくつかの固有のプロパティは、プロパティリストの"スケール"テーマで設定します:




このプロパティは進捗バー、ルーラー、数値ステッパー、およびダイアルタイプのインジケーターの場合に表示されます。またインジケータータイプにより設定可能なプロパティも異なります。以下これらの各プロパティについて説明します:

- **最小と最大**: インジケーターの最小値と最大値。
  数値ステッパーの場合これらの値は、時間タイプの場合秒を表し、日付タイプの場合は無視されます。
- **単位**: 目盛の表示単位。
- **ステップ**: 使用時に各値の間にあけることができる最小の間隔。
  数値ステッパーの場合このプロパティは、時間タイプの場合秒を、日付タイプの場合日数を表します。
- **ラベル位置**: ラベルが表示される際の位置。
  - **右揃え**: ラベルはインジケーターの右または下に表示されます。
    
    
  - **左揃え**: ラベルはインジケーターの左または上に表示されます。
    
    
- **ラベル**: ラベルを表示/非表示にします。
- **目盛**: ラベルの隣に目盛を表示、または非表示にします。
  
  

**オブジェクトメソッドの実行**

フォームエディターでインジケーターオブジェクトが選択されると、プロパティリストの"アクション"テーマには"**オブジェクトメソッド実行**"プロパティが追加されます。

このオプションを選択した場合、ユーザーがインジケーターの値を変更すると、On Data Changeイベントが生成され、オブジェクトメソッドが実行されます。デフォルトでは、変更後にメソッドが実行されます。

**プログラムからインジケーターを管理する**

インジケーターに関連付けた変数により、その表示を管理します。メソッドを用いて、この変数に値を代入したり、またはインジケーターの値を使用したりします。例えばフィールドまたは入力可オブジェクトのメソッドを使用して、サーモメーターを管理できます:

```
vTherm:=[Employees]Salary
```

このメソッドはSalaryフィールドの値を*vTherm*変数に代入します。このメソッドは例えばSalaryフィールドのオブジェクトメソッドとして記述できます。

逆にインジケーターを使用してフィールドの値を管理することも可能です。ユーザーはインジケーターをドラッグして値を設定します。このメソッドは次の通りです:

```
[Employees]Salary:=vTherm
```

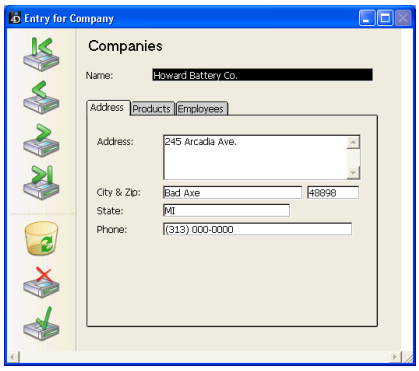
このメソッドはインジケーターの値をSalaryフィールドに代入します。ユーザーがインジケーターをドラッグすると、Salaryフィールドの値が変わります。

## タブコントロール

### 使用方法

タブコントロールは、ユーザーが一組の仮想画面の中から選択することのできるオブジェクトを作成します。この画面はタブコントロールオブジェクトにより囲まれています。タブをクリックして各画面にアクセスします。

次の複数ページフォームではタブコントロールオブジェクトが使用されています。

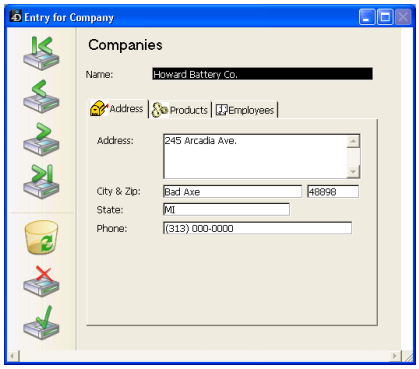



各画面を移動するには目的のタブをクリックします。

この画面は、マルチページフォームの各ページを表わしたり、またはユーザがタブがクリックすると変わるオブジェクトを表わすこともできます。タブコントロールをページ移動ツールとして使用する場合、ユーザがタブをクリックするとFORM GOTO PAGEコマンドまたはページ指定標準アクションが使用されます。

タブコントロールの他の利用法は、サブフォームやリストボックスに表示されるデータを制御することです。例えば名刺帳はタブコントロールを用いて実現することができます。タブにはひらがなの各文字を表示し、タブコントロールの動作としてはユーザがクリックした文字と一致するデータをロードします。

各タブにはラベルだけ、またはラベルと小さなアイコンを表示することができます。ラベルにアイコンが含まれる場合、そのアイコンは各ラベルの左側に表示されます。次の図はアイコンを使用するタブコントロールの例です:




タブコントロールを作成すると、4Dがタブの間隔と配置を管理します。 開発者は配列形式でラベルを定義するか、または階層リスト形式でアイコンとラベルを定義する必要があるだけです。

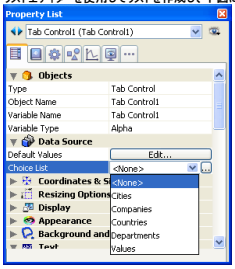
タブコントロールが十分大きく、ラベルとアイコンの双方が設定されたタブをすべて表示できる場合は、その両方が表示されます。タブコントロールの大きさが足らず、ラベルとアイコンを両方とも表示できない場合、4Dはアイコンだけを表示します。すべてのアイコンが収まりきらない場合、表示される最後のタブの右側にスクロール矢印が置かれます。このスクロール矢印を使用し、アイコンを左右にスクロールできます。

Mac OSの場合、タブコントロールを標準位置 (上) の他、左、右、または下に配置することができます ("タブ位置の変更 (Mac OSのみ)"参照)。

### タブコントロールにラベルを追加する

タブコントロールにラベルを設定するには、いくつかの方法があります:

- タブコントロールをデフォルト値のリストに関連付けることができます。このリストには、プロパティリストの"データソース"テーマの"デフォルト値"プロパティの横にある"**編集...**"ボタンからアクセスできます。この件に関する詳細はデフォルト値のリストを参照してください。デフォルト値は自動でテキスト配列にロードされます。タブコントロールに関連付けた変数名を用いてこの配列を参照することができます。
- リストエディターを使用してリストを作成し、下図に示すようにそのリストを選択リストとしてタブコントロールへ割り当てます:




  必要な場合は、リストエディターを使用して各リスト項目に小さなアイコンを関連付けることもできます。この件に関する詳細は項目にアイコンを設定するを参照してください。
- フォームの各ページの名前を格納するテキスト配列を作成することができます。このコードはユーザにフォームを表示する前に実行しなければなりません。例えばこのコードをタブコントロールのオブジェクトメソッドに置いて、On Loadイベントが生じたときにこのメソッドを実行します。

```
ARRAY TEXT(arrPages;3)
arrPages(1):="Name"
arrPages(2):="Address"
arrPages(3):="Notes"
```

**注:** ページの名前を階層リストに保存し、Load listコマンドを使用して値をロードすることも可能です。

### GOTO PAGEコマンド

タブコントロールのメソッドでFORM GOTO PAGEコマンドを使用できます:

```
FORM GOTO PAGE(arrPages)
```

On Clickedイベントが発生すると、このコマンドが実行されます。この後On Unloadイベントの発生時にこの配列をクリアします。

オブジェクトメソッドの例を次に示します:

```
Case of
   :(Form event=On Load)
      LIST TO ARRAY("Tab Labels";arrPages)
   :(Form event=On Clicked)
      FORM GOTO PAGE(arrPages)
   :(Form event=On Unload)
      CLEAR VARIABLE(arrPages)
End case
```

### ページ指定アクション

タブコントロールに**ページ指定**アクションを割り当てることができます。このアクションを選択すると、4Dはクリックされたタブコントロールの番号に相当するフォームのページを自動的に表示します。

例えば、ユーザーが3番目のタブコントロールをクリックすると、4Dはカレントフォームの3ページ目（存在する場合）を表示します。タブコントロールをプログラムから管理したい場合は"動作なし"を選択します。

標準アクションについては標準アクションを参照してください。

**タブ位置の変更 (Mac OSのみ)**

フォーム上のタブコントロールの位置を指定することができます。このプロパティにはすべてのプラットフォームからアクセスできますが、プラットフォームインタフェースが"システム"が指定されている場合にMac OS上でのみ動作します。タブコントロールは上（標準）、または下に配置することができます。

タブコントロールの位置はプロパティリストの"アピアランス"テーマで指定します。

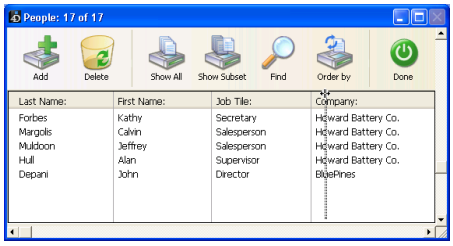
位置がカスタマイズされたタブコントロールをWindowsまたは"印刷"プラットフォームインタフェースを用いて表示すると、自動的に標準の位置（上）に戻されます。

## スプリッター

### 使用方法

スプリッターはフォームを2つのエリアに分割します。ユーザーはいずれかの方向へスプリッターを移動してエリアを拡げたり縮めたりすることができます。水平方向または垂直方向のスプリッターを作成できます。スプリッターでは各オブジェクトのサイズ調整プロパティが考慮されます。つまり作成するデータベースのインタフェースをすべてカスタマイズすることができます。また、スプリッターは"プッシャー"(押し込みタイプ)にすることも可能です。

通常スプリッターは列のサイズを変更できるよう、リストフォームで使用されます:




スプリッターの一般的な特徴をいくつか次に説明します:

- あらゆるタイプのフォーム上にスプリッターを必要なだけ設置可能であり、一つのフォーム上で水平と垂直のスプリッターを一緒に使用することができます。
- スプリッターはオブジェクトを横切ることができます(オーバーラップ)。スプリッターを動かすとこのオブジェクトのサイズが変更されます。
- フォーム上で移動されたオブジェクトが完全に表示されたままになるように、また別のスプリッターを超えないように、スプリッターの停止位置が計算されます。"以降のオブジェクトを移動する(プッシャー)"プロパティをスプリッターに割り当てると、スプリッターを右方向または下方向へ動かしても停止することはありません。
- スプリッターを使用するフォームのサイズを変更すると、フォームが表示されている間だけ、フォームの新しいサイズが保存されます。フォームを閉じると、最初の大きさに戻ります。

スプリッターは挿入されると線として表示されます。その境界スタイルを変更してさらに細い線に設定したり、カラーを変更したりすることができます。また**自動スプリッター**標準アクションを割り当てた透明ボタンをスプリッターとして使用することも可能です(標準アクション参照)。

### 隣接するオブジェクトのプロパティとの相互作用

フォーム上では、スプリッター周辺にある各オブジェクトのリサイズオプションに基づいて、スプリッターとこれらのオブジェクトとが作用し合います:

| オブジェクトのリサイズオプション | 水平スプリッターの上、または垂直スプリッターの左にあるオブジェクト(1) | 水平スプリッターの下、または垂直スプリッターの右にあるオブジェクト | |
|---|---|---|---|
| | | 非"プッシャー"スプリッター | "プッシャー"スプリッター |
| なし | そのまま変わらない | 次に停止するまでスプリッターとともに移動する(スプリッターとの相対的な位置は変更されない)。下または右への移動時の停止位置はウインドウの境界または別のスプリッター。 | 無制限にスプリッターとともに移動する(スプリッターとの相対的な位置は変更されない)。停止は行われない(次節を参照)。 |
| 拡大 | 元の位置のままだが、スプリッターの新しい位置に基づいてサイズが調整される | | |
| 移動 | スプリッターとともに移動 | | |

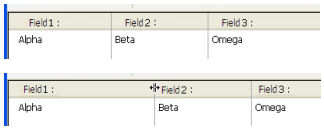
(1) この位置にあるオブジェクトを超えて、右側(水平)、または下側(垂直)へスプリッターをドラッグすることはできません。

**注:** スプリッターが定義される矩形内にすべて納まるオブジェクトは、スプリッターと一緒に移動します。

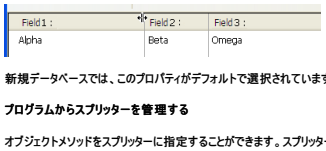
### 以降のオブジェクトを移動するプロパティ

プロパティリストの"リサイズオプション"テーマにおいて、スプリッターオブジェクトに対し**以降のオブジェクトを移動する**(プッシャー)プロパティを利用できます。スプリッターオブジェクトにこのプロパティを使用すると、そのオブジェクトの右側(垂直スプリッター)または下側(水平スプリッター)にある他のオブジェクトは、スプリッターと一緒に押し出されて移動し、停止していません。

"プッシャー"スプリッターを移動した場合の結果を次に示します:




スプリッターに対してこのプロパティを指定しない場合、結果は次のようになります:




新規データベースでは、このプロパティがデフォルトで選択されています。

### プログラムからスプリッターを管理する

オブジェクトメソッドをスプリッターに指定することができます。スプリッターを移動する間On Clickedイベントでこのメソッドが呼び出されます。

各スプリッターには倍長整数型の変数が割り当てられます。この変数はオブジェクトメソッドやフォームメソッドで使用することができます。この変数の値により、スプリッターの最初の位置に対する現在位置がピクセル単位で示されます。

- この値が負数の場合:スプリッターは上方向または左方向へ移動されました。
- この値が正数の場合:スプリッターは下方向または右方向へ移動されました。
- この値が0 の場合:スプリッターは元の位置に移動されました。

プログラムからスプリッターを移動することも可能です。これを行うには、関連付けた変数の値を設定するだけです。例えば、垂直スプリッターにsplit1という名前の変数が関連付けられている場合、split1:=-10 という命令を実行すると、ユーザが手動で動かす場合と同じように、スプリッターは10 ピクセル左方向へ移動します。

この命令文が記述されているフォームメソッドまたはオブジェクトメソッドを実行し終わると、実際に移動が行われます。

**自動スプリッター**アクション(標準アクション参照)を使用すると、独自のスプリッターをフォームに作成することができます。このアクションを透明ボタンタイプのオブジェクトに割り当てることができます。この標準アクションが透明ボタンに指定されると、ボタンはスプリッターと全く同じように機能します。例えば、この透明ボタンにピクチャをペーストすれば、自分のスプリッターのカスタムインタフェースを作成することができます。このタイプのボタンに関する詳細はボタンを参照してください。

# Webエリア

Webエリアはフォーム中で静的および動的なHTMLページ、ファイル、ピクチャー、JavaScript、Flash、PDF(1) 、さらにはMS Officeドキュメント (Windows環境でMS Officeがインストールされている場合) などのWeb用ファイルを表示することのできるオブジェクトです。.

(1) PDFドキュメントを表示するために、Mac OSはシステムに含まれるプレビューツールを使用しますが、WindowsではAcrobat Readerのインストールが必要です。

Webエリアの描画エンジンはアプリケーションの実行プラットフォームおよび**統合WebKitを使用**オプションの設定状態により異なります。

同じフォーム内に複数のWebエリアを配置できます。

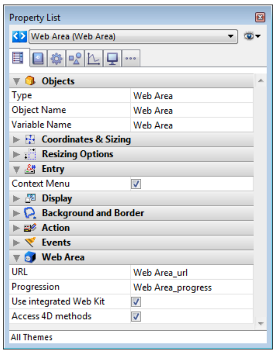
いくつかの標準アクション、ランゲージコマンド、そして汎用および専用のフォームイベントを使用して、開発者はWebエリアの動作を制御できます。特別な変数を使用してエリアと4D環境間で情報を交換できます。

この節ではフォームエディターにおけるWebエリアの利用や設定に関連する原則を説明します。このオブジェクトをプログラムで管理することに関する情報は、*4Dランゲージリファレンスマニュアル*のWebエリアを参照してください。

またWebエリアを使用する際にはいくつか特別なルールに従わなくてはならない点に留意してください。この点についてはWebエリアの利用に際する注意点で説明しています。

### 特別なプロパティ

フォームエディターでWebエリアを選択すると、プロパティリストには専用のプロパティが表示されます:




#### コンテキストメニュー

Webエリアで**コンテキストメニュー**プロパティ("入力"テーマ内)を設定すると、フォーム実行時にユーザーは標準のコンテキストメニューにアクセスできます。

メニューの内容はプラットフォームの描画エンジン (Mac OSではApple WebKit、WindowsではActiveX Web Controls) により設定されます。

**注:** WA SET PREFERENCEコマンドを使用してコンテキストメニューへのアクセスを制御できます。

#### 割り当てられる変数

Webエリアには2つの特別な変数“URL”と“進捗状況”が自動で割り当てられます。これらの変数はそれぞれWebエリアに表示されているURL やロード中のページのパーセンテージを知るために使用できます。デフォルトでこれらの変数はそれぞれWebエリア_urlとWebエリア _progressという名前です。必要に応じてこの名称を変更できます。

これらの変数はWebエリアを選択した状態でプロパティリストで設定できます。

これらの変数の動作に関する詳細は*4Dランゲージリファレンスマニュアル*の Webエリアのプログラムによる管理 を参照してください。

#### 統合されたWebKitを使用する

このオプションを使用して、Webエリアで使用する描画エンジンを2つのうちから選択することができます:

- **"統合されたWeb Kitを使用する" が選択されていない場合 (デフォルト):** この場合4Dはシステムの最適なエンジンを使用します。
  Windowsではマシンにインストールされた最新のInternet Explorerです。
  Mac OSではマシンにインストールされたWeb Kitです。
  この結果自動的にHTMLファイルやJavaScriptの最新Web描画エンジンを利用できることになります。しかしInternet ExplorerとWebKit間で若干描画に違いがでることがあります。
- **"統合されたWeb Kitを使用する"が選択されている場合:** この場合4Dは両プラットフォームでWebKitを使用します。WebKitはオープンソースのマルチプラットフォームWeb描画エンジンです。このエンジンは SafariやGoogle Chrome 等でも使用されています。
  4DにはWeb Kitが組み込まれています。この統合 WebKit を使用すれば、Web エリアの描画とその動作が ( ピクセル単位での若干の相違やネットワーク実装に関連する違いを除き ) プラットフォームに関わらず同じになります。
  このオプションが選択されると、OS により行われる自動更新などの利点を得ることができなくなります。使用される WebKit の新バージョンは4Dのリリースを通して提供されます。

**注:**

- Windows XPでは統合WebKitがサポートされていません。この場合標準モードが使用されます。
- 統合WebKit描画エンジンを使用する場合、以下の点に留意してください:
  - HTTPSプロトコルを使用してページを表示する場合、ページは描画されますが、認証チェーンは検証されません。
  - WindowsではPDFファイルは外部ウィンドウでのみ表示できます。

#### 4Dメソッドへのアクセス

**注:** このオプションは、**統合されたWeb Kitを使用**のオプションにチェックをしている場合のみ有効です。

このプロパティがチェックされている場合、特別なJavaScriptオブジェクト($4d)がWebエリア内に表示され、これを使用して4Dプロジェクトメソッドの呼び出しを管理することが出来るようになります。このオプションについての詳細な情報に関しては、4D *Language Reference* マニュアルの Webエリアのプログラムによる管理 セクションを参照して下さい。

### 標準アクション

Webエリアを自動で管理するために、4つの特別な自動アクション、**Open Back URL**、**Open Next URL**、**Refresh Current URL**、そして**Stop Loading URL**を使用できます。これらのアクションをボタンやメニューコマンドに割り当て、基礎的なWebインターフェースを素早く実装できます。これらのアクションについては標準アクションで説明しています。

### イベントとランゲージコマンド

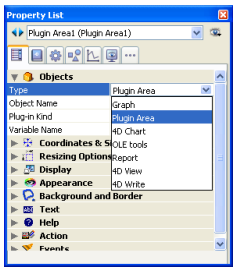
フォームイベントとランゲージコマンドを使用してWebエリアを制御できます。この点については4D ランゲージリファレンスマニュアルのWebエリアで説明しています。

## プラグインエリア

**プラグインエリア**は4Dプラグインによりすべてが制御されるフォーム上のエリアです。データベースを開く際、4Dはデータベースにインストールされたプラグインのリストを内部的に作成します。フォームにプラグインエリアを挿入すると、4Dは利用可能なプラグインをプロパティリスト上に一覧表示します。デフォルトでOLE tools（Windows のみ）、Reportを使用できます。

**注：** 一部のプラグインはフォームやプラグインウインドウで使用することができません。プラグインがフォームで使用できない場合、そのプラグインはプロパティリストのプラグインリストに表示されません。

エリアで使用するプラグインを指定するには、"オブジェクト"テーマの"タイプ"リストからその名前を選択します:




**注：** 選択されたオブジェクトタイプが"プラグインエリア"である場合、そのプラグインの作成者が詳細オプションを提供していると、**詳細設定**ボタンが使用可能になることがあります。この場合ボタンをクリックして、そのオプションを設定することができます。この詳細設定オプションのダイアログボックスはプラグインの制作元が制御するため、詳細設定オプションに関する情報はそのプラグインの製作者から提供されます。

作成したプラグインエリアが小さすぎる場合、4Dはエリアをボタンとして表示し、タイトルにはエリアに割り当てられた変数名が使用されます。実行時ユーザーはこのボタンをクリックしてプラグインを表示するためのウィンドウを開くことができます。

### プラグインのインストール

プラグインを4D環境にインストールするには、まず4Dを終了する必要があります。4Dは起動時にプラグインをロードします。プラグインのインストールに関する詳細はプラグインやコンポーネントのインストールを参照してください。

### プラグインの利用

フォームにプラグインエリアを追加できることで、カスタムアプリケーションを作成する際の可能性が限りなく広がります。プラグインではフォーム上にデジタル時計を表示するような単純な処理や、完全に機能するワープロ、スプレッドシート、グラフィック機能などを提供するなどの複雑な処理を実行できます。

これらの機能の大半は、4Dプラグインの形で4D開発環境から既に利用することができます。例えば次のようなプラグインがあります:

- 4D Internet Commands: 低レベルのネットワーク通信をサポートするためのツール
- 4D Write: ワープロアプリケーション
- 4D View: スプレッドシートおよびリスト管理アプリケーション

これらのプラグインに関する情報はそれぞれのドキュメントを参照してください。4D Doc Centerではこれらのプラグインのランゲージリファレンスを提供しています: 4D Internet Commands, 4D Write および 4D View

注: グラフを作成、表示するために使用される4D Chartプラグインも利用できます。しかしこのプラグインは4D v13において廃止予定ステータスになっています (4D Chart参照)。

独自にプラグインを作成したい場合、オープンソースのプラグイン制作キットを使用することができます。このキットの入手およびプラグイン作成に関する情報はhttp://sources.4d.com/trac/4d_4dpluginapi/ を参照してください。

### 組み込みのプラグインエリア

デフォルトで4Dは次のタイプの組み込みプラグインを提供しています:

- **グラフ**: グラフエリアはGRAPHと_o_GRAPH TABLEコマンドを使用してプログラムで管理できます。
- **OLE tools** (Windowsのみ): Windowsにおいて、OLEエリアを使用して、データベース内に他のアプリケーションのドキュメントを含むウィンドウを表示できます。この点についてはOLE Toolsマニュアルで説明しています。
  **注：オブジェクト**メニューの**OLEオブジェクト挿入**コマンドを使用して直接OLEエリアを作成することもできます。
- **Report**: Reportエリアを使用すると、フォーム中にクイックレポートを挿入できます。クイックレポートには表形式でデータが表示され、サマリーの計算等が行えます。このエリアは独自のメニューバーや4Dランゲージコマンドを使用して制御できます。詳細はクイックレポートを参照してください。

## アクティブオブジェクトのプロパティ


- オブジェクト名
- データ入力の制御と補助
- 表示フォーマット
- 関連付け(値または参照番号)
- ドラッグ＆ドロップ
- 標準アクション
- ミラー効果 (Windows)
- マルチスタイル (リッチテキストエリア)
- 複数行
- スクロールバー
- コンテキストメニュー (ピクチャー)
- キーボードショートカット
- ウィンドウの配置の記憶

# オブジェクト名

## オブジェクト名と変数名

各アクティブオブジェクトにはオブジェクト名と変数名が関連付けられています。変数名とオブジェクト名は違っていてもかまいません。同じフォーム内で複数のアクティブオブジェクトに同じ変数名を割り当てることができますが、オブジェクト名はそれぞれユニークでなければなりません。

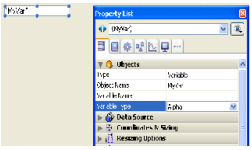
**注:** オブジェクト名と変数名の上限は31文字です。

4Dランゲージを使用する際、変数名やオブジェクト名いずれかを使用してアクティブフォームオブジェクトを参照できます (詳細については4Dランゲージリファレンスのオブジェクトプロパティを参照)。

フォームオブジェクトと変数の命名規則については4Dランゲージリファレンスのテーブルとフィールドの命名規則と識別子を参照してください。

### ダイナミック変数

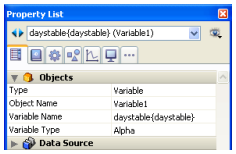
(ボタン、入力可変数、チェックボックス等の)フォームオブジェクトに割り当てられる変数を、必要に応じて動的に、4Dに作成させることができます。これを行うには、オブジェクトのプロパティリストで"変数名"フィールドを空にします:



詳細は4Dランゲージリファレンスの変数を参照してください。

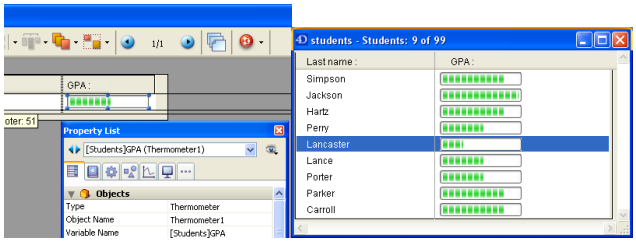
## 変数名に式を使用する

フォームオブジェクトの変数名に、変数名ではなく、値を返す式を指定できます:




有効な4D式であれば何でも受け入れられます。シンプルな式、フォーミュラ、4D関数、プロジェクトメソッド名、標準の[Table]Fieldシンタックスを使用したフィールド名を使用できます。式はフォームが実行されたときに評価され、フォームイベント毎に再評価されます。当然この場合、変数の値をプログラムを使用して管理することはできません。

この方法を使用して多くの可能性が生まれます。例えばリストフォームで進捗インジケーターに数値フィールドを割り当て、グラフィカルに値を表示できます:




**注:** 入力された式が変数名とメソッド名両方で使用されている場合、4Dはメソッド名が指定されたものと判断します。

## データ入力の制御と補助

プロパティリストを使用して、アクティブオブジェクトへのデータ入力を制御または補助する様々なプロパティを設定できます。これらのプロパティを使用すると以下のことが可能です:

- 選択とデータ入力に関係する属性を設定できます。
- 入力可能な文字を制御するための入力フィルターを設定できます。
- プレースホルダーテキストを表示させることが出来ます。
- 選択リストを割り当てることができます。
- 指定リストや除外リストを指定できます。
- 最大値や最小値、デフォルト値を設定できます。
- オブジェクトの選択モードを指定できます。
- スペルチェックができます。
- ヘルプメッセージを表示できます。

### 入力可

**入力可**属性はユーザーがオブジェクトに値を入力できるかどうかを指定します。これはフィールドおよび変数オブジェクトのプロパティリストの"入力"テーマで設定します。

フォームウィザードの**リレートフィールド入力可**チェックボックスの選択を解除している場合、リレートしたテーブルのフィールドは入力ができません。この場合、プロパティリストで入力可にチェックを入れると、そのフィールドオブジェクトへの入力を許可できます。

すべてのアクティブオブジェクトはデフォルトで入力可です。特定のフォーム上の特定のフィールドやオブジェクトを入力不可にしたい場合、入力可チェックボックスの選択を解除します。入力不可のオブジェクトはデータの表示のみを行います。メソッドでフィールド名や変数名を使用してデータを制御します。入力不可オブジェクトでもOn Clicked, On Double Clicked, On Drag Over, On Drop, On Getting Focus そして On Losing Focus フォームイベントは使用できます。これを利用すれば、カスタムコンテキストメニューを容易に管理し、ドラッグ＆ドロップ可能なインターフェースをデザイン、そして入力不可変数を選択することができます。

**注:**

- プロパティリストの内容はコンテキストを考慮します。フィールドや変数の入力可属性が選択解除されると、入力制御に関連するプロパティ (必須入力、タブ有効、入力フィルター等) はリストに表示されなくなります。
- ストラクチャーエディターでフィールドに対し表示のみプロパティを設定できます (UNREGISTER CLIENT参照)。この場合**入力可**オプションはこのフィールドのプロパティリストに表示されません。

### 必須入力

**必須入力**チェックボックスを選択すると、そのフォーム上でフィールドが必須入力となります。フィールドに値が入力されない場合、4Dはレコードを受け入れません。

デフォルトでフィールドは必須入力ではありません。すべてのフォームでフィールドを必須入力にするには、ストラクチャーエディターのインスペクターウィンドウで必須入力属性を設定します。特定のフォームのみでフィールドを必須入力にしたい場合、オブジェクトプロパティの必須入力オプションを選択します。

### 入力可/必須入力とフィールドプロパティ

入力可と必須入力属性は、ストラクチャーエディターで設定するフィールド属性と同じです (フィールドプロパティ参照)。これらの属性を特定のフォームで異なるものに設定したい場合、フィールドごとにプロパティで設定できます。これらの属性はフィールドおよび変数オブジェクトのプロパティリストの"入力"テーマで設定できます。

これらの属性はストラクチャーエディターで設定したフィールド属性を上書きしません。ストラクチャーエディターですでにフィールドに対し表示のみ属性が割り当てられていると、入力可フォーム属性を使用してそれを入力可にすることはできません。ストラクチャーエディターでフィールドに対しすでに必須入力属性が設定されている場合、必須入力フォーム属性を選択解除して非必須入力にすることはできません。入力可と必須入力チェックボックスは、ストラクチャーエディターでの設定を反映しません。

### タブ有効

フォーム上のアクティブオブジェクトごとに**タブ有効**属性を設定できます。この属性が選択されると、オブジェクトは入力順に組み込まれ、ユーザーが**Tab**キーを押すことで選択されるようになります。

デフォルトでこの属性は、この属性をサポートするすべてのアクティブオブジェクトに対し選択されています。選択を解除すると、そのオブジェクトは入力順から外されます。しかしマウスクリックやメソッドによって選択することは可能です。

入力順についてはデータの入力順を変更するを参照してください。

入力ができない (ボタンなどの) オブジェクトに対し、**タブ有効**プロパティを設定できます。この場合Tabキーを使用してオブジェクトを選択することができますが、依然として入力はできません。

**タブ有効**プロパティは、オブジェクトがフォーカスを得ることが可能な場合にのみ有効です。しかしいくつかのオブジェクトはフォーカスを得ることができてもタブが有効でない場合があります (例えばクリックで選択可能だがタブでは選択できないオブジェクト)。この場合、オブジェクトはデータ入力順には含められません。

### フォーカス表示

このプロパティはアクティブオブジェクト (入力可か不可かにかかわらず) や入力できないフィールドのプロパティリスト中、"入力"テーマで設定できます。

オブジェクトに対し**フォーカス表示**プロパティが選択されていると、そのオブジェクトはフォーカスを得ることができ、またキーボードなどを使用してアクティブ化することができます。オブジェクトはフォーカスを得ると、オブジェクトごとあるいはOSごとに定められた方法でハイライトします。ただし"フォーカスの四角を隠す"オプションが選択されている場合を除きます (後述)。

チェックボックスにフォーカスがある



チェックボックスが選択されているが、フォーカスが表示されない状態

入力できないオブジェクトに**フォーカス表示**プロパティが設定されていると、ユーザーはエリアの内容を選択、コピー、およびドラッグ＆ドロップすることができます。

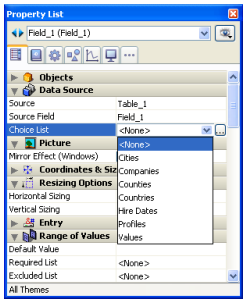
### フォーカスの四角を隠す

実行時、フィールドやすべての入力可オブジェクトは (Tabキーやクリックにより) フォーカスを得たときグラフィカルにそれを表現します。**フォーカスの四角を隠す**オプションをチェックするとこのグラフィカル表現を無効にできます。

### 選択リスト

入力可フォームオブジェクトには選択リストを割り当てることができます。3タイプの選択リスト、標準の選択リスト、指定リスト、除外リストがあります。

標準の選択リストはプロパティリストの"データソース"テーマで、指定リストや除外リストは"値の範囲"テーマで指定します。




選択リストを割り当てる前に、リストエディターを使用して事前にリストを作成しなければなりません (リスト参照)。

#### データソース選択リスト

"データソース"テーマの選択リストドロップダウンリストを使用してフィールドに対しデータソース選択リストを割り当てると、4Dはデータ入力中に 自動で選択リストを表示します。

デフォルトで、選択リストはデータ入力時に自動的にウィンドウに表示されます。この選択リストは実行されているフォームにおいてフィールドまたは入力可能オブジェクトが選択されたときに表示されます。そしてユーザーはリストから入力する値を選択できます。

また、ポップアップメニュー、ドロップダウンリストと コンボボックスを使用して選択リストを関連づけ、フィールドまたは変数に対しての値の入力と表示を管理することが出来ます(選択リストが階層リストだった場合、最初のレベルのみが使用可能です)。これをするためには、プロパティリストの"変数名"の中にフィールドまたは変数の名前を入力するだけです(ポップアップメニュー、ドロップダウンリストを参照して下さい)。

選択リストから選択した値は入力によって上書きすることが出来ます(リストが指定リストであった場合を除く)。

#### 除外リスト

除外リストを使用すると、リスト中の項目は入力できなくなります。例えば従業員に使用される入力フォーム上のフィールドに、管理者だけが入力できる値のリストを除外リストとして割り当てることができます。

#### フィールドと選択リスト

フィールドの選択リストはテーブルまたはフォームレベルで設定できます。テーブルレベルで選択リストを割り当てるにはストラクチャーエディターのインスペクターウィンドウを使用します ([#title id="3200"/]参照)。するとこのフィールドにはすべてのフォームとデータベースの検索エディターで選択リストが有効になります。フォームレベルで 選択リストを割り当てると、そのフォームだけで選択リストが有効になります。この場合、ポップアップメニュー、ドロップダウンリスト または コンボボックス を使用することによってフィールド内のデータ入力とデータ表示を直接管理できます。

#### 指定リスト

指定リスト ("値の範囲"テーマ) は有効な値のリストを指定するために使用します。例えば指定リストとして役職名のリストを指定できます。こうすると、事前に作成されたリスト中の役職名だけ有効な値となります。

指定リストを指定しても、フィールドが選択されたときに自動でリストは表示されません。フォーム実行中に指定リストの内容を表示したい場合は、"データソース"テーマの選択リストドロップダウンで同じリストを指定します。

### 入力フィルター

**日本語利用時の注意点**: *入力フィルターは日本語IMEと互換性がありません。入力文字種の制限及び幅を使用した入力文字数の制限もできません。例えば半角数字のみ2文字だけ入力を許可する目的で、入力フィルターに&9##と指定しても、IME経由での全角数字やその他日本語文字の入力を防ぐことはできませんし、任意の数の文字が入力できてしまいます。アプリケーション仕様としてこのような制御が必要な場合は4Dコマンドを使用する必要があります。*

入 力フィルターを使用するとデータ入力中にユーザーがタイプできる文字を制御できます。先に説明したデータ入力制御とは異なり、入力フィルターは文字ごとに 処理が行われます。例えばパーツ番号が常に2つの文字とそれに続く3つの数字で構成されるとき、そのパターンを入力フィルターを通して強制することができ ます。さらに特定の文字や数字のみを使用するよう制御することもできます。

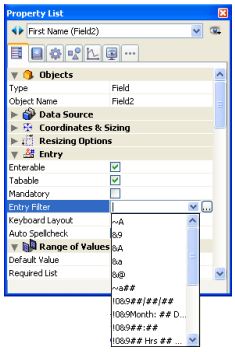
入力フィルターはデータ入力時にのみ動作します。ユーザーがオブ ジェクトの選択を解除した後のデータ表示には効果がありません。通常入力フィルターと表示フォーマットを一 緒に使用します。フィルターはデータ入力を制約し、表示フォーマットはデータ入力後の値の表示を制御します。

データ入力中、入力フィルターはタイプされるたびに文字を評価します。ユーザーが無効な文字をタイプすると (例えば数字ではなく文字)、4Dは入力を受け付けません。ユーザーが有効な文字を入力するまで値は変更されません。

#### 入力フィルターを選択する

入力フィルターは入力フィルタードロップダウンリストから組み込みのフィルターを選択する、またはエリアに入力フィルターコードをタイプすることで設定できます。入力フィルタードロップダウンリストには日付、時間、または文字オブジェクトのフィルターが含まれます。

ほとんどの場合、4D組み込みのフィルターを使用することができます。しかしカスタマイズされたフィルターを作成することも可能です (フィルターエディター参照)。開発者が作成したフィルターの名前は縦棒 (|) 付きで入力フィルターリストの先頭に追加されます。




ドロップダウンリストから選択した後に入力フィルターを更新することができます。例えば大文字と小文字、記号およびワイルドカード文字 (@) の入力を許可したい場合、まず以下のフィルターを選択します:
&"a-z;0-9; ;,;.;-"
そして以下のように変更します:
&"a-z;0-9; ;,;.;~;@"

入力フィルターのコードは通常アンパサンド (&) から始まります。これは4Dに、この後に続く文字が入力フィルターであることを伝えます。コードがチルダ (~) から始まる場合、これは "&" と同じ意味を持ちますが、文字が自動で大文字に変換される点が異なります。
&は通常"A"、"a"、あるいは"9"が続きます。それぞれ大文字のみを許可 (A)、大文字と小文字を許可 (a)、数字のみを許可 (9) を意味します。例えば&9は数字のみ、&Aは大文字のみを許可します。シャープ (#) は許可される文字数を指定するために使用します。フィルターに#が含まれない場合、フィルターは文字数を制限しません。例えば&9は任意の数の数字入力を許可します。フィルター&9##は2つの数字入力を受け入れます。
エクスクラメーションマーク (!) は、ユーザーが入力できる文字数を示す記号を変更するために使用します。!が使用されない場合、4Dは下線 (_) をユーザーが入力可能な場所に表示します。例えば!?&9##は2つの?を入力可能な場所に表示し、かつその場所には数字のみを受け入れます。入力フィルターの作成についてはフィルターとフォーマットのコードを参照してください。

#### 組み込み入力フィルターの説明

入力フィルタードロップダウンリストから選択できる入力フィルターの説明は以下の表の通りです:

| 入力フィルター | 説明 |
|---|---|
| ~A | すべての文字が入力可能、大文字に変換される。 |
| &9 | 数字だけが入力可能。 |
| &A | 大文字の文字だけが入力可能。 |
| &a | 文字だけが入力可能(大文字と小文字)。 |
| &@ | 特殊記号を除く数字と文字が入力可能。 |
| ~a## | 2桁の任意の文字が入力可能、大文字に変換される |
| !0&9##.##.## | 標準の日付入力フォーマット。入力領域に0を表示し、任意の数値を入力可能。 |
| !0&9####.##.## | 標準の日付入力フォーマット。入力領域に0を表示し、任意の数値を入力可能。 |
| !0&9##年##月##日 | カスタム日付入力フォーマット。入力領域に0を表示し、任意の数値を入力可能。 |
| !0&9####年##月##日 | カスタム日付入力フォーマット。入力領域に0を表示し、任意の数値を入力可能。 |
| !0&9##:## | 時間入力フォーマット。時と分だけを入力可能。入力領域に0を表示し、任意の4桁の数値を入力。 |
| !0&9##:##:## | 時間入力フォーマット。時、分、秒を入力。入力領域に0を表示し、任意の6桁の数値を入力。 |
| !0&9##時##分##秒 | 時分秒の文字付きの時間入力フォーマット。入力領域に0を表示し、任意の数値を入力可能。各単位の前に2桁の数値を入力。 |
| !0&9####-#### | 入力領域に0を表示。任意の数値を入力可能。4桁と4桁の数字をハイフンで区分。 |
| !_&9(##) !0###-#### | 先頭の2桁の入力領域にアンダースコアを表示しカッコで囲み、残りの入力領域に0を表示。 |
| !0&9###-###-#### | 入力領域に0を表示。任意の数値を入力可能。3桁と3桁と4桁の数字をハイフンで区分。 |
| !0&9###-##-### | 入力領域に0を表示。任意の数値を入力可能。3桁と2桁と4桁の数字をハイフンで区分。 |
| ~"A-Z;0-9; ;,;.;-" | 大文字の文字と句読点。大文字の文字、数字、スペース、コンマ、ピリオド、ハイフンだけを入力可能。 |
| &"a-z;0-9; ;,;.;-" | 大文字と小文字の文字と句読点。小文字の文字、数字、スペース、コンマ、ピリオド、ハイフンだけを入力可能。 |
| &"0-9;.;-" | 数値。数字、小数点、ハイフン(マイナス記号)だけを入力可能。 |

#### 入力フィルターと表示フォーマットを併用する

入力フィルターの使用時には通常、対応する表示フォーマットを使用します。入力フィルターが動作するのはデータ入力時のみです。そのフィールドから外に出てしまったら、データ表示についての効力は失います。例えばソーシャルセキュリティ番号の入力フィルター(&9###-##-####)を使用した場合、これに対応するソーシャルセキュリティ番号の表示フォーマット(###-##-####)を選択すべきです。表示フォーマットが選択されていない場合、ハイフンなしの番号だけがそのフィールドに表示されます。

よく使用されるフィールドタイプ用の入力フィルターとそれに対応する表示フォーマットを以下に紹介します:

| フィールドタイプ | 入力フィルター | 表示フォーマット |
|---|---|---|
| 州 | ~a## | (none needed) |
| 郵便番号 (標準) | &9##### | (必要なし) |
| 郵便番号 (拡張) | &9#####-#### | #####-#### |
| 電話番号 | &9###-#### | ###-#### |
| | &9(###) ###-#### | (###) ###-#### |
| | &9 ###-###-#### | ###-###-#### |
| 日付 | !0&9##/##/## | (任意の日付フォーマット) |
| 時間 | !0&9##:## | (任意の時間フォーマット) |

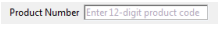
表示フォーマットは詳細フォーム、一覧フォーム、クイックレポート上で使用できます。クイックレポートでの表示フォーマットの利用については表示フォー

マットの設定を参照してください。

**プレースホルダーテキスト**

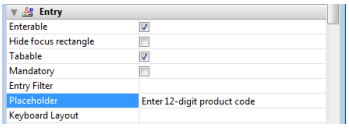
4D では、フォームのフィールド内にプレースホルダーテキストを表示することができます。

このテキストはフィールド内で半透明のテキストとして表示され、入力されるデータに関するヘルプ、指示、具体例などを表示します。このテキストはユーザーが文字をエリアに入力した瞬間に表示されなくなります:



プレースホルダーテキストは、フィールドの中身が消去されると再び表示されます。

プレースホルダーテキストは、プロパティリストの入力のテーマから設定することが出来ます:




**プレースホルダー**のオプションは以下のオブジェクトに対して設定可能です:

- 変数
- フィールド
- コンボボックス

プレースホルダーとして表示できるデータの型は以下の通りです:

- 文字列(テキストまたは文字フィールド)
- 日付または時刻(**ヌルのときブランク**にするのプロパティがチェックされている場合に限る)

xliff 参照を ":xliff:resname" の形でプレースホルダーとして使用することもできます。例えば、

```
:xliff:PH_Lastname
```

参照を使用する場合にはプレースホルダーのフィールドには参照のみを渡して下さい。参照と静的なテキストを組み合わせることはできません。

**注:** プレースホルダーのテキストは、新しい OBJECT SET PLACEHOLDER と OBJECT Get placeholder コマンドを使用するとプログラミングによって設定したり取得したりすることができます。

**最大値と最小値**

オブジェクトプロパティの"値の範囲"テーマにある入力エリアで最大値と最小値を指定して、数値や日付時間の範囲を制限できます。

データ入力時、ユーザーが最小値より小さい値や最大値より大きい値を入力すると、警告メッセージが表示されます。4Dは入力カーソルをそのエリアに戻し、ユーザーは有効な値を入力しなければなりません。

最大値や最小値を設定するには、制限値として指定する値を入力します。制限を設定する入力エリアには適切な入力フィルターを設定します。例えば日付フィールドに最大値と最小値を設定する場合、日付入力フォーマットを設定します。

設定した値は上限と下限になります。つまりユーザーは設定された最大値と最小値を入力することができます。最小値より小さい値、あるいは最大値よりも大きい値が拒否されます。例えば最大値が15のとき、ユーザーは15を入力できますが、16は入力できません。ユーザーが入力できる値を制限するためにメソッドを使用することもできます。メソッドではより正確で有用な情報をユーザーにフィードバックできます。またデータベースの他の値に基づく制限値を設定することもできます。例えばメソッドを使用すれば顧客ごとの与信限度額を検証できます。

指定リストを使用して非連続の許可値のリストを作成することもできます。

詳細については先の指定リストセクションおよびリストの作成と変更を参照してください。

**デフォルト値**

フィールドや入力可能オブジェクトにデフォルト値を割り当てることができます。デフォルト値は新規レコードが作成され、最初に表示されるときに代入されます。エリアが入力不可に設定されていなければ、デフォルト値を書き換えることができます。

プロパティリストの"値の範囲"テーマ内、デフォルト値欄に、デフォルト値として指定する値をタイプします。デフォルト値はフィールド型に対応していなければなりません。日付、時刻、シーケンス番号については、4Dが提供する記号を使用することができます。日付と時刻はシステムから取得されます。シーケンス番号は4Dが自動で生成します。自動で使用できるデフォルト値の記号は以下の通りです:

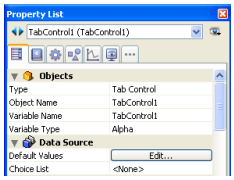
| 記号 | 意味 |
|---|---|
| #D | 本日の日付 |
| #H | 現在の時刻 |
| #N | シーケンス番号 |

レコード毎のユニーク番号を生成するためにシーケンス番号を使用できます。シーケンス番号は倍長整数型で新規レコード毎に生成されます。番号は1から始まり、1づつ増加します。シーケンス番号が割り当てられたレコードが削除されても、その番号は再利用されません。シーケンス番号はテーブルごとに管理されます。詳細は*4Dランゲージリファレンス*のSequence numberコマンドを参照してください。

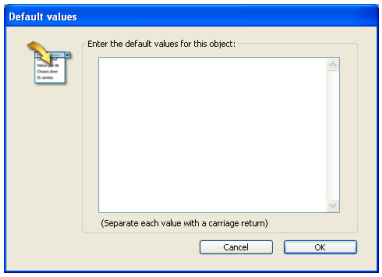
**デフォルト値のリスト**

(コンボボックス、スクロールエリア、ポップアップメニュー、タブコントロール、ドロップダウンリストなど) オブジェクトが値のリストを表示する場合、デフォルト値として使用する値のリストを定義できます。リストはフォームが表示される前にオブジェクトにロードされます。

デフォルト値のリストを受け入れるオブジェクトの場合、デフォルト値エリアがボタンになります:




このボタンをクリックすると、デフォルト値ダイアログボックスが表示されます:




デフォルト値のリストを入力します。値は行ごとに分割されます。**OK**をクリックするとデフォルト値ダイアログボックスが閉じられ、プロパティリストウィンドウに戻ります。

デフォルト値ダイアログボックスにデフォルト値を入力すると、リストは自動で、オブジェクトに割り当てられた変数と同じ名前の配列にロードされます。ランゲージを使用してこの配列を参照することで、オブジェクトを管理できます。

**注:**

- リストエディターで作成したリストを使用して、階層リストやタブコントロールのデフォルト値を設定することもできます。
- メソッドを使用してデフォルト値を設定することもできます。オブジェクトにメソッドでデフォルト値を設定するためには、オブジェクトあるいはフォームメソッドをOn Loadイベントで実行するように設定し、そこで値を割り当てます。
- リストを受け入れるオブジェクトでは、リストエディターを使用してデフォルト値リストを作成し、Load listコマンドで階層リストを作成することができます。その後"階層リスト"テーマのコマンドを使用してリストで作業を行うことができます。リストはOn LoadフォームイベントやOn Startupデータベースメソッド等でロードできます。

**キーボードレイアウト**

このオプションはフィールドまたは入力可のオブジェクトに対して特定のキーボードレイアウトを関連付けます。例えば、国際的なアプリケーションに対し

て、フォーム内にギリシャ文字で入力しなければならないフィールドがあった場合、"ギリシャ語"のキーボードレイアウトをこのフィールドに対して関連付けることができます。これにより、このフィールドがフォーカスを受けている場合にはデータ入力時にキーボード設定が自動的に変わります。

デフォルト値<None>のとき、オブジェクトはカレントのキーボードレイアウトを使用します。

**注意:** このオプションは、OBJECT SET KEYBOARD LAYOUT と OBJECT Get keyboard layout コマンドを使用して動的に設定することが可能です。

**スペルチェック**

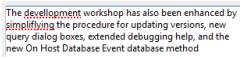
4Dにはカスタマイズ可能な、統合されたスペルチェッカーが含まれています。文字やテキスト型のフィールドや変数、および4D Writeエリアでチェックを有効にできます。

**自動スペルチェック**プロパティ ("入力"テーマ) を使用して、そのオブジェクトのスペルチェックを有効にします。有効にすると、スペルチェックはデータ入力時に自動的に実行されます。

**注:** オブジェクトごとにSPELL CHECKINGコマンドを使用してチェックを行うこともできます。

**スペルチェックのプロセス**

4Dでは、スペルチェックは絶えずテキストエリア内で実行されていて、エラーは破線の下線が直接テキストに引かれてハイライトされます:



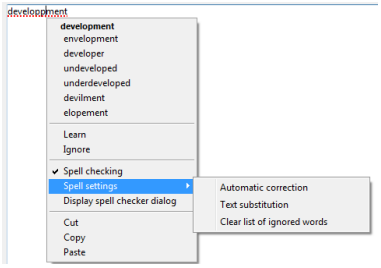

異なる色の破線はエラーの種類を表しています:

- 赤の破線は綴りの間違いを表示します
- 緑の破線は文法の間違いを表示します(OS Xのシステムのスペルチェッカーのみ)
- 青の破線は置き換える単語を表示します(**置き換えテキストを表示**のオプションがチェックされている必要があります。以下を参照して下さい)

ユーザーをそれをみて、コンテキストメニュー(下線の引かれた単語を右クリックで表示)またはスペルチェッカーダイアログを使用して修正することが出来ます。

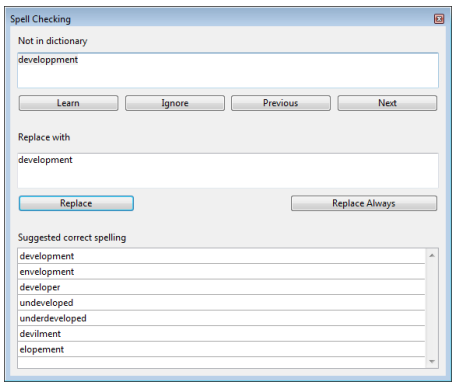
**コンテキストメニューまたはスペルチェッカーダイアログボックス**

スペルチェックのオプションを表示するためには、下線の引かれた単語を右クリックします。コンテキストメニューが表示され、その中にスペルチェッカーのコマンドがあります:




**注: 自動スペルチェックとコンテキストメニュー**のオプションがチェックされている必要があります。

**スペルチェッカーダイアログ表示**のオプションを選択すると、修正した値を入力するダイアログが表示されます:

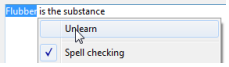



このダイアログボックスは、セッションの間、ユーザーが閉じるまでアプリケーション内の全てのウィンドウに表示されるフローティングするウィンドウです。

**スペルチェッカー機能**

4Dのスペルチェッカーでは修正候補の提示をしてくれるだけではなく、以下のオプションと機能が用意されています(コンテキストメニュー内またはスペルチェッカーダイアログボックス内):

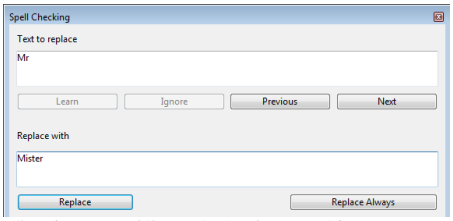
- **登録**: この未知の単語を辞書に登録し、今後スペルチェッカーによって検知されることはなくなります。
  - OS X 環境下では、学習した単語は /Users/[*UserName*]/Library/Spelling の場所に恒久的に保存されています(ここで学習した単語は、システムスペルチェッカーを使用する他のアプリケーションでも表示されなくなります)。
  - ハンスペル辞書では、学習した単語はカレントのアプリケーションデータのユーザーシステムディレクトリ内のカスタムの辞書内に保存され、メインの辞書がロードされる際に同時に呼び出されます(旧バージョンと同様の動作です)。
- **登録解除**(コンテキストメニュー): このオプションは、以前に学習した単語を選択したときに表示されます。選択すると、この単語を学習した単語のリストから除外し、その単語は再びスペルチェッカーによってエラーとして表示されるようになります。




- **無視**: この未知の単語はそのままの状態で下線が表示されないようになります。ただし、その後再び検知されたときには下線が表示されます。スペルチェッカーは、ドキュメントごとに無視する単語の表を保存しています。**無視した単語の一覧を消去**のオプションを選択することで一覧表を削除することができます(以下を参照して下さい)。
- **前へ/次へ**(ダイアログボックス): この未知の単語は何も変更されず、下線を引かれたままで、スペルチェッカーはテキスト内の次の未知の単語を検索します。
- **スペルチェック**(コンテキストメニュー): カレントのプロセスにおいて、エリア内のスペルチェッキングを全体的に有効化または無効化します。
- **置換**(ダイアログボックス): テキスト内で選択された単語が最初の入力フィールドにある単語と一致する場合、二番目のフィールド内の単語で置換され、同時にスペルチェッカーは次のエラーを検索します。
- **常に置換**(ダイアログボックス): **置換**と同じですが、その置換は記憶されます(以下の**テキストの置換**を参照して下さい)。

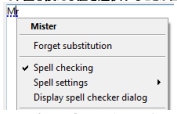
**スペルチェック**のサブメニューでは、以下の様な機能が使用可能です:

- **スペルを自動的に修正**: カレントのプロセス内において、自動修正モードを有効化もしくは無効化します。このモードが有効化されている場合においては、未知の単語は最も近い既知の単語で自動的に置き換えられます(ただしあまりに曖昧な場合は行われません)。修正は、入力の際に行われます。
  デフォルトでは、このオプションは無効になっています。
- **テキストの置換**: テキストの置換を有効化または無効化します。この機能は、ある単語を他の単語で置き換えるというものです。例えば、"Mr." を "Mister" で置き換える、といったようなことができます。"置き換えられる単語/以下の単語で置き換え"の単語のペアを作成するためには、スペルチェッカーダイアログを使用する必要があります。"辞書にありません" エリアに置き換えたい単語を入力し、それを置き換える単語を "以下で置換" エリアに入力して下さい。その後**常に置換**ボタンをクリックします:




置換はアプリケーションの中全体において実行されます。ハンスペル辞書においては、スペルチェッカーがアプリケーション内の置き換えを記録する表を、アプリケーションのユーザーシステムディレクトリ内に保存します。OS Xのシステムスペルチェッカーの場合、この表はシステムの置換と統合されます(システム環境設定言語とテキスト内の"記号とテキストの置換を使用"オプション)。
置き換えのプロセスを使用するためには、**スペル設定** 内の**テキストの置き換えを表示**のオプションがチェックされている必要があります(以下を参照して下さい)。

- **テキストの置き換えを表示**(このオプションは、**テキストの置換**のオプションがチェックされている場合にのみ表示されます): このオプションがチェックされているとき、スペルチェッカーは置き換える必要があり得る単語について青の下線を引きます。ユーザーはその単語を右クリックすることによって、置き換える値を選択することが出来ます。

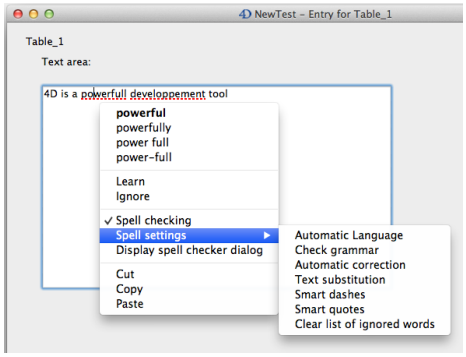



このオプションがチェックされていないとき、置換はユーザーの操作なしに自動的に行われます。

- **無視した単語の一覧をクリア**: ドキュメントの中で無視するよう指定された単語の一覧を消去します。

#### OS X のシステムのスペルチェッカーにおけるその他のオプション

OS X のシステムのスペルチェッカーには、他にもいくつかの修正のためのオプションが存在します:




- **言語ごとに自動**: 文章の内容をもとにして、文章の言語と使用すべき辞書を自動的に検知します。
デフォルトでは、ハンスペルのスペルチェッカーと同じように4Dアプリケーションの言語を使用します。
- **文法をチェック**: 文章の文法のチェックを有効化します。
- **スマートダッシュ記号**: 入力中、ハイフン二つ(--)をダッシュ(—)で置き換えます。
- **スマート引用符**: 引用符を、カレントの言語に合わせたスマート引用符へと置き換えます。

#### 辞書を管理する

4Dは、スペルチェックの際に使用される辞書は以下の通りです:

- WindowsとOS X 両方の環境下では、オープンソースのハンスペル辞書が使用できます (4D *Language Reference* マニュアルのHunspell辞書のサポートを参照して下さい)。
- OS X環境下では、システムのスペルチェッカーも使用できます。
4D v14 は、OS X 環境下では、デフォルトとしてシステムのスペルチェッカーを使用します。[#cmd id="642"/] コマンドを使用してハンスペルのスペルチェックを使用することもできます。OS Xなお、OS X のシステムのスペルチェッカーにも新機能が追加されています(以下の項目を参照して下さい)。

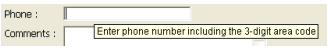
デフォルトでは、4Dはカレントのアプリケーション言語に対応した辞書を使用します。しかし SPELL SET CURRENT DICTIONARY コマンドを使用して、別言語の辞書を強制的に開くことができます。4Dではスペルチェックが受け入れるカスタマイズされた単語のリストも含め、ランゲージ ごとに特別な辞書を設定し使用することができます。これらの辞書の設定については付録 D: 特別な辞書を使用するを参照してください。

### ヘルプメッセージ

フォーム上のフィールドやアクティブオブジェクトにヘルプメッセージを追加して、ユーザーがデータベースを使いやすくし、生産性が向上するよう手助けすることができます。ヘルプメッセージはTipsとして表示されます。

例えばデータ入力時に日付は年、月、日をスラッシュで区切るようユーザーに指示するヘルプメッセージを作成できます。

ヘルプTipはマウスがフィールドやオブジェクトに重なった時に表示されます:




**注:** フィールドにはデータベーストラクチャーレベルでヘルプメッセージを結び付けることができます。この場合フィールドのヘルプTipは、そのフィールドが置かれたすべてのフォームで表示されます。詳細はUNREGISTER CLIENTの"ヘルプTip"を参照してください。同じフィールドに2つの異なるヘルプTipが結び付けられている場合、フォームレベルのヘルプが優先されます。

プロパティリストにある"ヘルプ"テーマのヘルプTipドロップダウンリストを使用してフォーム中の任意のオブジェクトにヘルプTipを結び付けることができます。以下のいずれかが可能です:

- 既存のヘルプTipを選択する。ヘルプTipは事前にヘルプTipsエディターで定義されていなければなりません。
- エリアに直接ヘルプメッセージを入力する。この場合XLIFFアーキテクチャーを利用することができます。ここにXLIFF参照を入力し、アプリケーション言語に対応するメッセージを表示させられます。XLIFFに関する詳細は付録 C: XLIFFアーキテクチャーを参照してください。4D参照を使用することもできます (スタティックテキスト中で参照を使用する参照)。

選択あるいは入力したメッセージは、フォーム中で選択されたフィールドやオブジェクト用にヘルプTipとして表示されます。

選択したオブジェクトのヘルプTip割り当てを削除するには、プロパティリストで**なし**を選択します。

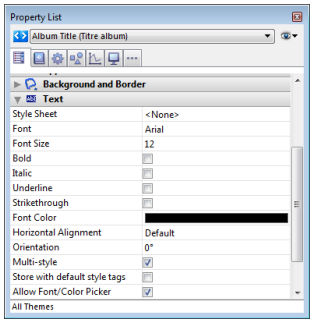
### 選択を常に表示

このプロパティは文字やテキスト型のフィールドや変数で使用できます。このプロパティを選択すると、オブジェクト中で選択した文字列の反転状態が、フォーカスを失った後も表示されるようになります。これにより、テキストスタイルを更新するようなインターフェースの実装が容易になります (マルチスタイル(リッチテキストエリア)参照)。

### カラーピッカーとフォントピッカーの使用を許可

OPEN FONT PICKER コマンドとOPEN COLOR PICKERコマンドを使用することによってカラーピッカーとフォントピッカーを表示させることが出来ます。ユーザーは、これらのウィンドウをクリックすることにより、フォーカスされているオブジェクトのカラーやフォントを直接変更することが出来ます。

これらのユーザーの行動を管理するために、この機能を有効化するためには「テキスト」テーマ内の**ピッカーの使用を許可**のプロパティにチェックをします。



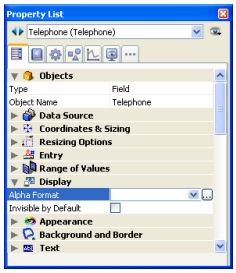

このプロパティは、フィールド、変数、そしてコンボボックス型のオブジェクトに対して存在します。初期設定では全てのフォームオブジェクトにおいてこのチェックは外れています。ピッカーウィンドウを使用してカラーやフォントを変更可能にしたいオブジェクト全てについて、明示的にこのプロパティにチェックをする必要があります。

## 表示フォーマット

4Dが提供する表示フォーマットを使用すると、画面表示や印刷に関する選択の幅が広がります。表示フォーマットは、フィールドおよび入力可エリアや入力不可エリア(変数)の双方に適用することができます。フィールド内容の表示に用いるフォーマットは、4Dに格納される実際の値に影響を与えません。

フィールドの表示フォーマットをフォームごとに変えることができます。例えばある値を入力フォームでは円記号(¥)なしで表示し、出力フォームでは円記号付きで表示したい場合もあるでしょう。

表示フォーマットはプロパティリストの"表示"テーマで設定します。




*コンボボックスを使用して表示フォーマットを表示・選択したり、直接入力したりできます。[...]ボタンをクリックするとフィルター設定ウィンドウが開きます* (フィルターエディター参照)。

選択するフィールドや変数の型に応じて、この選択コンボボックスには異なるフォーマットが表示されます。

**注:** プロパティリストの"オブジェクト"テーマにある**変数タイプ**プロパティを使用して、変数のタイプを指定します。

指定した型の組み込みフォーマットは常に表示されます。ツールボックスのフィルタ/フォーマットエディターを用いて追加された表示フォーマットは、名前の先頭に縦棒 "¦" が付けられてリストの先頭に表示されます (フィルターエディター参照)。

**注:** フォームエディターにおいて**オブジェクト**メニューから**フォーマット表示**を選択すると、オブジェクトのラベルの代わりに表示フォーマットを表示できます。

### 日付フォーマット

日付フォーマットは、表示や印刷時に日付を表示する方法を制御します。データ入力の際は選択した表示フォーマットとは関係なく、YYYY/MM/DD形式で日付を入力します。

**注:** 数値フォーマットや文字フォーマットと異なり、日付表示フォーマットは4Dの組み込みフォーマットのなかから選択しなければなりません。

利用可能な日付表示フォーマットは以下のとおりです:

| フォーマット | 例 |
|---|---|
| System date short | 1999/03/25 |
| System date abbreviated | 1999年3月25日 |
| System date long | 1999年3月25日 |
| Internal date short special | 99/03/25 *(1)* |
| Internal date long | March 25, 1999 |
| Internal date abbreviated | Mar 25, 1999 |
| Internal date short | 1999/03/25 |
| ISO Date Time*(2)* | 1999-03-25T00:00:00 |

*(1)* 年は、1930年~2029年の間は2桁の数字で表示されますが、それ以外の場合は4桁で表示されます。これはデフォルト設定ですが、SET DEFAULT CENTURYコマンドで変更することができます。
*(2)* ISO Date TimeフォーマットはXMLの日付と時間表現の標準 (ISO8601) に対応します。これは主にXMLフォーマットやWebサービスのデータを読み込んだり書き出したりするために使用します。

**注:** 表示フォーマットにかかわらず、年度を2 桁で入力すると、4Dは年が00~29の間であれば21 世紀とみなし、30~99の間であれば20 世紀とみなします。これはデフォルト設定ですが、SET DEFAULT CENTURYコマンドで変更することができます。

**注**: 実際の表示はシステム設定によりこの例とは異なります。

#### ヌルの時ブランクにする

デフォルトでヌルの日付は"00/00/00"と表示されます。プロパティリストの"表示"テーマにある**ヌルの時ブランクにする**オプションを選択すると、日付がヌルの場合やSQLのNULL属性を含む場合に、エリアは空で表示されます。

### 時間フォーマット

時間フォーマットは、表示や印刷時に時間を表示する方法を制御します。選択した表示フォーマットとは関係なく、データ入力の際は24時間制の"HH:MM:SS"フォーマット、または12時間制の"HH:MM:SS AM/PM"フォーマットで時間を入力します。

**注:** 文字や数値の表示フォーマットとは異なり、時間の表示フォーマットはフォーマットポップアップメニューから選択しなければなりません。

次の表は、時間フィールドの表示フォーマットとそれぞれのフォーマットの例を示しています:

| フォーマット | コメント | 04:30:25の時の例 |
|---|---|---|
| HH:MM:SS | | 04:30:25 |
| HH:MM | | 04:30 |
| Hour Min Sec | | 4 時 30 分 25 秒 |
| Hour Min | | 4 時 30 分 |
| HH:MM AM/PM | | 4:30 AM |
| MM SS | 00:00:00からの経過時間 | 270:25 |
| Min Sec | 00:00:00からの経過時間 | 270 分 25 秒 |
| ISO Date Time | 時間に関連するXML標準表現に対応。主にXMLフォーマットでのデータのやり取りに使用します。 | 0000-00-00T04:30:25 |
| System time short | システムに定義された標準の時間フォーマット | 4:30:25 |
| System time long abbreviated | *Mac OSのみ*: システムに定義された時間フォーマットの短縮型。WindowsではSystem time shortフォーマットと同じ | 4:30:25 |
| System time long | *Mac OSのみ*: システムに定義された時間フォーマット。WindowsではSystem time shortフォーマットと同じ | 04:30:25JST |

**注**: 実際の表示はシステム設定によりこの例とは異なります。

#### ヌルの時ブランクにする

デフォルトでヌルの時間は"00:00:00"と表示されます (実際の表示は表示フォーマットにより異なります)。プロパティリストの"表示"テーマにある**ヌルの時ブランクにする**オプションを選択すると、時間がヌルの場合やSQLのNULL属性を含む場合に、エリアは空で表示されます。

### 数値フォーマット

**注**: 数値フィールドには整数、倍長整数、整数64bit、実数、そしてフロート型が含まれます。

数値フォーマットは表示や印刷時に数値を表示する方法を制御します。選択した表示フォーマットとは関係なく、データ入力の際は数値だけを (必要に応じ小数点やマイナス記号も) 入力します。

4Dはプロパティリストの"表示"テーマで様々な数値表示フォーマットを提供しています。ポップアップメニューからフォーマットを選択するか、コンボボックス内に直接入力できます。またツールボックスのフィルターとフォーマットエディターで設定したカスタムフォーマットを選択することもできます。この場合そのフォーマットをオブジェクトプロパティ内で変更することはできません。エディターにはコンボボックス右側の[...]ボタンからアクセスできます。

カスタムフォーマット (およびフィルター) 名は数値および文字フォーマットリストの先頭に、縦棒 (¦) の後に表示されます。

#### プレースホルダー

それぞれの数値表示フォーマットでは、数値記号(#)、ゼロ(0)、キャレット(^)、アスタリスク(*)をプレースホルダとして使用します。表示しようとする各桁に対して1つのプレースホルダを使用し、独自の数値表示フォーマットを作成できます。

例えば3桁の数字を表示する場合、###というフォーマットを使用できます。フォーマットにより許可された桁数を超えて入力すると、4Dは<<<をフィールドに表示し、表示フォーマットで指定された桁数を超える入力が行われたことを示します。

ユーザーがマイナスの数値を入力すると、左端の文字はマイナス記号として表示されます(負数の表示フォーマットが指定されていない場合)。##0というフォーマットであれば、マイナス26は-26と表示されます。マイナス260は<<<と表示されますが、これはプレースホルダが3桁分しか指定されていないところに、マイナス記号により1つのプレースホルダが使用されてしまい、桁あふれしたためです。

**注:** 表示フォーマットとは関係なく、4Dはフィールドに入力された数値を受け入れ、保存します。データが失われることはありません。

各プレースホルダ文字は、先行のゼロや末尾のゼロを表示する上で、その効果に違いがあります。先行のゼロとは小数点より左側の数値の先頭にあるゼロのことです。末尾のゼロは小数点より右側の数値の終わりにあるゼロのことです。

例えば##0というフォーマットを使用して3桁の数字を表示するものとします。ユーザーがフィールドに何も入力しないと、フィールドには0が表示されます。26と入力すると、フィールドには26と表示されます。

次の表は、先行のゼロや末尾のゼロに関する各プレースホルダの効果を表わしています:

| プレースホルダー | 千項及び末尾のゼロ |
|---|---|
| # | 何も表示しない |
| 0 | 0を表示 |
| ^ | スペースを表示 (1) |
| * | アスタリスクを表示 |

(1) キャレット (^) はほとんどのフォントの数字の幅と同じ幅を占めるスペースを生成します。

#### 区切り文字

数値表示フォーマット (科学的記数法を除く) は自動でシステムの地域パラメターに基づきます。4DはOSに定義された小数点と千の位区切り文字を使用して "." と "," 文字をそれぞれ置き換えます。0や#に続くピリオドとコンマはプレースホルダー文字として扱われます。

**互換性に関する注意:** バージョン11よりも前の4Dでは、数値表示フォーマットにシステムの地域パラメターは考慮されませんでした。変換されたデータベースでは互換性オプションを使用してこの動作をコントロールできます (互換性ページ参照)。

#### 小数点とその他の表示文字

表示フォーマット内では1つの小数点を使用することができます。ユーザが小数点を入力するかどうかに関係なく小数点を表示したい場合、ゼロの間に小数点を置かなければなりません。

フォーマット内で他の文字を使用することもできます。文字を単独で使用したりプレースホルダーの前後に配置すると、その文字が常に表示されます。例えば次のようなフォーマットの場合:

￥##0

円記号はプレースホルダーの前に置かれているため、常に表示されます。

文字がプレースホルダーの間に置かれている場合、両側に数字が表示される場合のみ、その文字が表示されます。例えばフォーマットを次のように指定したとき:

###.##0

コンマは、ユーザが少なくとも4桁以上の数値を入力した場合にのみ表示されます。

数値表示フォーマットにおいて、スペースは文字として扱われます。

#### 正数、負数、ゼロのフォーマット

数値表示フォーマットは最大で3つの部分に分けられ、それぞれ正数、負数、ゼロの値に対応する表示フォーマットを指定できます。それぞれのパートは以下のように並び、セミコロンで区切られます:

正数;負数;ゼロ

3つの部分すべてを指定する必要はありません。1つの部分だけを使用する場合、4Dはすべての数値に対してそのフォーマットを使用し、負の数の先頭にマイナス記号を配置します。

2つの部分を使用する場合、4D は1番目のフォーマットを正数とゼロに対して使用し、負数には2番目のフォーマットを使用します。3つの部分を使用すると、1番目のフォーマットを正数、2 番目を負数、3 番目をゼロに使用します。

**注**: 3番目の部分(ゼロ)は解釈されず、文字の置き換えを行いません。###;###;#と指定した場合、ゼロ値は#と表示されます。言い換えると、表示フォーマットとして実際に指定されたものが、ゼロ値として表示されます。

次の数値表示フォーマットの例は、円記号とカンマを表示し、負の数値はカッコ内に入れ、ゼロを表示しません:

￥###,##0.00;(￥###,##0.00);

2つ目のセミコロンにより、ゼロの表示には何も使用しないことを4Dに指示している点に注目してください。次のフォーマットは前の例と似ていますが、2つ目のセミコロンが指定されていません。これにより、ゼロに対して正数のフォーマットを使用するよう4Dに指示しています:

￥###,##0.00;(￥###,##0.00)

この場合、ゼロは"￥0.00"と表示されます。

#### 科学的記数法

科学的記数法で数値を表示したい場合アンパサンド (&) に続けて表示したい桁数を指定します。例えば次のフォーマットを指定すると:

&3

759.62は以下のように表示されます:

7.60e+2

科学的記数法フォーマットは、表示される数値を自動的に丸める唯一のフォーマットです。前述の例では、数値が7.59e+2 と切り捨てられずに7.60e+2 に丸められている点に注意してください。

#### 16進フォーマット

次の表示フォーマットを使用して、数値を16進表記で表示することができます:

- &x: このフォーマットでは16進数が"0xFFFF"形式で表示されます。
- &$: このフォーマットでは16進数が"$FFFF"形式で表示されます。

#### XML記法

&xmlフォーマットを使用すると、数字をXML標準ルールに沿ったものにします。とくに小数点がポイント (ピリオド) に変換されます。

#### 数値を時間として表示する

"&/"の後に数字を指定することにより、数値を時間として（時間フォーマット使用して）表示することができます。時間は午前0 時を基点とした秒数として計算されます。フォーマット内の数字は表示フォーマットドロップダウンメニュー上でその時間フォーマットが表示される順番に相当します。

例えば次のフォーマットは:

&/5

ドロップダウンメニューの5番目の時間フォーマット（AM/PM で表わす時間）に対応します。このフォーマットが指定された数値フィールドの場合、25000は次のように表示されます:

6:56 AM

#### 例題

次の表は各種フォーマットの数値表示への効果を表わしています。正数、負数、ゼロという3つの欄では1234.50、-1234.50、0 がそれぞれどのように表示されるかを示しています。

| 入力されたフォーマット | 正数 | 負数 | ゼロ |
|---|---|---|---|
| ### | <<< | <<< | |
| #### | 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 |
| $*,**0.00;–$*,**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 |
| &xml | 1234.5 | -1234.5 | 0 |

### 文字フォーマット

文字フォーマットは、表示や印刷時に文字フィールドや変数にデータを表示する方法を制御します。以下は文字フィールド用に提供されるフォーマットのリストです:

```
###-####
(##) ####-####
##-####-####
####-##-####
00000
```

デフォルトでプロパティリストの"表示"エリアで文字フォーマットが利用可能です。このリストからフォーマットを選択するか、コンボボックスにタイプすることができます。フォーマットポップアップメニューには主に使用される文字フォーマット (電話番号等) が用意されています。またツールボックスのフィルターとフォーマットで設定したカスタムフォーマットを選択することもできます。この場合そのフォーマットをオブジェクトプロパティで変更することはできません。
フォーマットコンボボックスの右にある[...]ボタンをクリックすると、フォーマットエディターにアクセスできます。
開発者が作成したカスタムフォーマットやフィルターは先頭に縦棒 (|) が追加され、リストの先頭に表示されます。

シャープ (#) は文字表示フォーマットのプレースホルダーです。ハイフンやスペース、その他の句読点を表示したい場所に挿入できます。表示したい実際の句読点と、文字データを表示する場所には#を置きます。

例えば部品番号外貨のようなフォーマットの時:

RB-1762-1

文字フォーマットを以下のように書けます:

##-####-#

ユーザーが"RB17621"と入力すると、フィールドには以下の通りに表示されます:

RB-1762-1

フィールドに実際に格納される値は"RB17621"です。

フォーマットが許可するよりも多くの文字が入力されると、4Dは最後の文字を表示します。例えばフォーマットが以下の時:

(#######)

そしてユーザーが"proportion"と入力すると、フィールドには以下のように表示されます:

(portion)

フィールドには実際"proportion"が格納されます。表示フォーマットにかかわらず、4Dは入力された文字を受け入れ、格納します。情報は失われません。

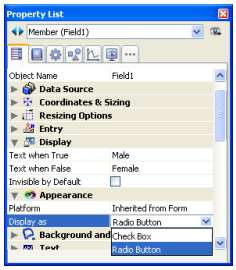
### ブールフォーマット

ブールフィールドには2値 (TrueとFalse) のうちひとつを格納することができます。ブールフィールドはラジオボタンのペアまたは1つのチェックボックスとして表示できます。

フィールドプロパティでブールフィールドの表示形式とラベルを指定できます。ボタンやチェックボックスだけを表示し、ラベルは表示したくない場合は、フォーム中のフィールドラベルを削除します。

#### ラジオボタン

ブールフィールドのアピアランスは、アピアランステーマの表示形式ドロップダウンリストを用いて指定します。このリストからラジオボタンを選択すると、表示テーマの下にある"テキスト(True 時)"と"テキスト(False 時)"入力エリアに各値のラベルを入力することができます。




次に示す図のように、フォーム上でボタンは横並びに表示されます。

先頭の1 文字が異なるラベルを使用する場合は、データ入力時に最初の1 文字を入力することにより、ラジオボタンを選択することができます。例えばラベルが"Male"と"Female"であれば、フィールドの選択時に"M"キーを押してMale を選択し、"F"キーを押してFemale を選択できます (日本語文字は対応していません)。

フィールドにデータが保存される際、次のルールが適用されます。1番目のボタンが選択されるとフィールドはTrueになります。2番目のボタンが選択されるとフィールドはFalseになります。デフォルトでフィールドはFalseになっています。

#### チェックボックス

"アピアランス"テーマの"表示形式"からチェックボックスを選択します。このオプションを選択すると、"オブジェクト"テーマに"タイトル"というラベルがついた入力エリアが表示されます。ここにチェックボックスのラベルを入力します。デフォルトのラベルは"Yes"です。

チェックボックスをブール型フィールドの表示形式として使用した場合、チェックボックスが選択されていればTrue、チェックボックスが選択されていなければFalseです。フィールドはデフォルトでFalseです。
フォーム実行中、フィールドはチェックボックスとして表示されます:




タイトルエリアにスペースを入力し、境界線スタイルを透過にすると、ラベルなしのチェックボックスを作成できます。この場合、チェックボックスのラベルは別なオブジェクトとして追加でき、四角を書いたり、動的参照 (スタティックテキスト中で参照を使用する参照)を用いたり、好みの位置に配置したりできます。以下の例ではテキストオブジェクトの前面にブールフィールドを配置しています。

### ピクチャーフォーマット

ピクチャーフォーマットはピクチャーが表示あるいは印刷される際の表示方法を制御します。データ入力時はフォーマットに関わらず、ユーザーはクリップ

ボードからのペーストやドラッグ＆ドロップでピクチャーを入力します。ピクチャーフォーマットはプロパティリストの"表示"テーマにあります。

トランケートとスケーリングオプションを選択しても、ピクチャーが変更されることはありません。ピクチャー表示フォーマットはピクチャーの表示にのみ影響します。

**スケーリング**

スケーリングを選択すると、ピクチャーはフィールドエリアの大きさに合うようにリサイズされます。

**トランケート (中央合わせ/中央合わせしない)**

トランケート (中央合わせ) フォーマットを選択すると、4Dはフィールドの中央にピクチャーを配置し、フィールドエリアに収まらない部分はエリアからはみ出します。

トランケート (中央合わせしない) フォーマットを選択すると、4Dはピクチャーの左上角をフィールドの左上角に合わせて配置し、フィールドエリアに収まらない部分はエリアからはみ出します。

**注:** ピクチャーフォーマットがトランケート (中央合わせしない) の場合、フィールドや変数エリアにスクロールバーを追加できます。詳細はスクロールバーを参照してください。

*トランケート (中央合わせ)*　　*トランケート (中央合わせしない)*

**スケーリング (プロポーショナル) とスケーリング (中央合わせ・プロポーショナル)**

スケーリング (プロポーショナル) を使用すると、ピクチャーエリアに収まるよう、比率を保ったままサイズが調整されます。スケーリング (中央合わせ・プロポーショナル) オプションも同様ですが、ピクチャーはエリアの中央に配置されます。

ピクチャーがエリアよりも小さい場合、サイズは変更されません。ピクチャーがエリアよりも大きい場合、そのエリア内に全体が表示されるよう、比率を保ったままサイズが小さくなります。

中央合わせを選択した場合、画像はエリアの中央に配置されます:

*スケーリング (プロポーショナル)*　　*スケーリング (中央合わせ・プロポーショナル)*

**バックグラウンド**

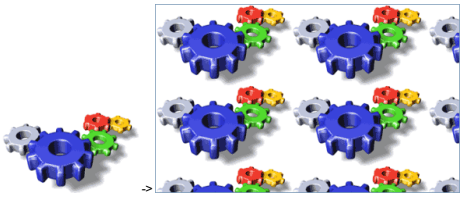
バックグラウンドはピクチャーを透過にします。画像の後ろに配置されたフィールドや変数などのオブジェクトは画像を通してみることができます。

**注:** バックグラウンドフォーマットのピクチャーを印刷すると、ビットマップとして印刷されます。

ピクチャーフィールドにこのフォーマットを適用すると、ユーザーはピクチャーをドラッグして、エリア内でピクチャーを移動できます。4Dはオブジェクトの位置を記憶します。

**繰り返し**

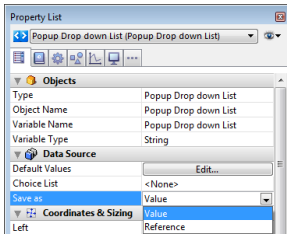
繰り返しフォーマットを持つピクチャーを含むエリアが拡大されると、ピクチャーは変形されず、エリア全体を埋めるのに必要なだけピクチャーが繰り返されます。




フィールドがオリジナルのピクチャーよりも小さいサイズにされた場合、ピクチャーはトランケート (中央合わせなし) されます。

## 関連付け(値または参照番号)

フィールドおよび変数オブジェクトとポップアップメニュー、ドロップダウンリストの"データソース"テーマの中に、関連付けというオプションがあります:




このオプションは、選択リストに関連付けされたフィールドと変数において、フィールドに保存する中身の型を指定します:

- **リスト項目の値**(デフォルトのオプション): ユーザーによって選択された項目の値が直接保存されます。例えば、ユーザーが"Blue"という値を選択した場合、この値がフィールドに保存されます。
- **リスト項目の参照番号**: 選択リスト項目の参照がオブジェクトに保存されます。この参照番号とは APPEND TO LIST または SET LIST ITEM コマンドの itemRef 引数、またはリストエディターを通してそれぞれの項目と関連付けされた数値です(項目に参照を追加する を参照して下さい)。

このオプションにより、メモリーを節約することが出来ます。フィールドに数値を保存するのは文字列を保存するより容量が軽いからです。またこれによりアプリケーションの翻訳が簡単になります。同じ項目の参照値を持つ、異なる言語で書かれた複数のリストを用意しておいて、アプリケーションの言語に応じたリストをロードするだけで済むようになるからです。

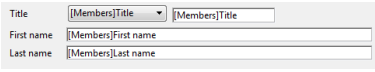
**リスト項目の参照番号**の使用の際には、以下の点に注意する必要があります:

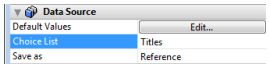
- 参照を保存するためには、フィールドまたは変数は、数値型である必要があります(リスト内に表示されている値の型とは関係ありません)。
- リストの項目には有効かつ固有の参照が関連付けられている必要があります。
- ポップアップメニューにおいてこのオプションを使用した場合、ポップアップメニューは配列ではなく、フィールドまたは変数と関連付けられている必要があります(選択リストの使用を参照して下さい)。
- このオプションは、ストラクチャの中で定義された選択リストにも有効です。この場合は、列挙されたフィールドが使用されているそれぞれのフォームにおいてこのオプションを選択してください。

**例**

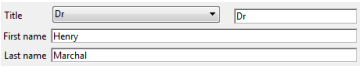
人を区別しやすくするために"Title"というフィールドを使用する場合を考えます。Mr, Ms だけでなく、Dr, Mgr, Hon, なども含めます。このためには、まず"Title"という名前の倍長整数型のフィールドを作成します。次に考えられ得る項目を全て含んだ"Title"という名の選択リストを定義し、それをフィールドと関連付けます。

入力フォームには、フォームの仕組みを説明するために"Title"のフィールドを二回表示するとします。一度目はポップアップメニューとして、もう一度は入力エリアとして。どちらのオブジェクトも同じ選択リストに関連付けられており、データは参照として保存されています:

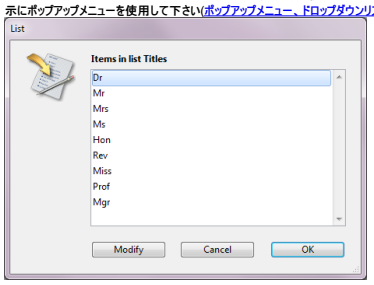


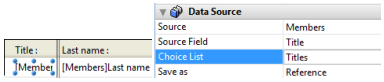

入力中、ポップアップメニューの値を選択すると、両方のオブジェクトに正しく表示されます:



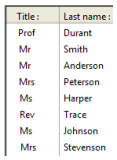

**注:** このフォーム内において、入力エリアにフォーカスが移ると、選択リストウィンドウが表示されます。このウィンドウを表示させないためには、入力と表示にポップアップメニューを使用して下さい(ポップアップメニュー、ドロップダウンリストを参照して下さい)。




同じやり方で、出力フォームを設定することもできます。"Title"フィールド内の関連付けのオプションでリスト項目の参照番号を選択して下さい:




実行すると、値は正常に表示されます。



| Title : | Last name : |
|---|---|
| Prof | Durant |
| Mr | Smith |
| Mr | Anderson |
| Mrs | Peterson |
| Ms | Harper |
| Rev | Trace |
| Ms | Johnson |
| Mrs | Stevenson |

## ドラッグ＆ドロップ

プロパティリストの"アクション"テーマにあるプロパティを使用して、フォーム内でのドラッグ＆ドロップを設定することができます。設定できるプロパティの数と種類はオブジェクトタイプにより異なります。

**ドラッグ可とドロップ可**

これらのプロパティはユーザーがオブジェクトをドラッグできるかどうか、およびユーザーがドラッグしたデータをオブジェクトが受け取ることができるかどうかを制御します。

特定のオブジェクトのドラッグやドロップを有効にしたい場合、対応するプロパティを有効にする必要があります。この場合、フォーム実行中にオブジェクトに対して行われたドラッグやドロップ処理は、対応するフォームイベントを生成します。開発者はメソッドを使用してドラッグ＆ドロップアクションを管理しなければなりません。詳細は4Dランゲージリファレンスマニュアルの[ドラッグ＆ドロップ]を参照してください。

**注：** 入力不可オブジェクトにドロップ可プロパティを割り当てることができます。この方法を使用すれば、入力不可に設定したフィールドや変数にドロップされた際の応答をプログラムで制御できます。

**自動ドラッグ＆ドロップ**

テキストオブジェクト (フィールド、変数、コンボボックス、およびリストボックス) やピクチャーオブジェクトは自動ドラッグ＆ドロップを使用できます。自動ドラッグ＆ドロップはテキストやピクチャーを直接、あるフォームエリアから他のフォームエリアに、ポイント/クリック操作でコピーや移動できます。これは同じ4Dエリア間、2つの4Dエリア間、あるいは4Dと (メモ帳など) 他のアプリケーション間で利用できます。

例えば自動ドラッグ＆ドロップを使用して、プログラムを使用せず、2つのフィールド間で値をコピーできます:

自動ドラッグ＆ドロップを有効にするには、そのオブジェクトのプロパティリスト中、"アクション"テーマで対応するプロパティを選択します。

詳細は4Dランゲージリファレンスマニュアルの[ドラッグ＆ドロップ]を参照してください。

## 標準アクション

この節では、ボタンや特定のアクティブフォームオブジェクト (またはメニューコマンド、メニューアクションを指定する参照) に割り当てることのできる各標準アクションについて説明します。

- **動作なし**: 標準アクションを実行しないボタンには、**動作なし**ボタンを使用します。メソッドを作成してボタンを管理する必要がある場合には、**動作なし**を選択します。例えばカスタムアプリケーションにおいて、独自の検索ダイアログボックスを表示するボタンには、**動作なし**標準アクションを指定します。これはメソッドを作成してカスタムダイアログボックスを表示しなければならないためです。
- **入力**および**キャンセル**: **入力**動作のボタンをクリックするとレコードが保存されます。これによりOn Saving New Record EventやOn Saving Existing Record Event、On Validateイベントが起動されます。**キャンセル**動作のボタンをクリックすると変更内容を保存せずにカレントレコードを終了します。
- **レコード削除**: **レコード削除**動作のボタンをクリックすると、削除の確認を求める警告が表示されます。警告ダイアログボックスで**OK**をクリックするとカレン トレコードが削除されます。サブレコードの入力フォームを使用している場合、**レコード削除**動作のボタンによりカレントサブレコードが削除されます。**レコード削除**ボタンをクリックした後、4Dは自動で出力フォー ムに戻ります。
  **注**: 新規レコードを追加している場合、この動作が指定されたボタンは自動的に選択不可になります。
- レコードナビゲーション動作: **次レコード**、**前レコード**、**先頭レコード**、**最終レコード**の各ボタンは、まずカレントレコードを保存してから、指定されたレコードをカレントレコードにします。レコードの並び順に従って、これらのボタンにより特定のレコードがカレントレコードに設定されます。
  サブレコードを入力している場合、これらのボタンはサブレコードに適用されます。
  このタイプのボタンは、そのボタンの動作が適切ではない場合には自動的に無効になります。例えば先頭のレコードが表示されている場合、**先頭レコード**と**前レコード**ボタンは使用不可になります。
- ページナビゲーション動作: **次ページ**、**前ページ**、**先頭ページ**、**最終ページ**の各ボタンは、マルチページフォームの該当するページを表示します。1 ページし か存在しない場合、これらのボタンは使用不可になります。
  このタイプのボタンは、そのボタンの動作が適切ではない場合には自動的に無効になります。例えば最初のページが表示されている場合、**先頭ページ**と**前ページ**ボタンは使用不可 になります。
- **サブレコード編集**: このアクションは3種類の状況下で利用できます。MODIFY SELECTION / DISPLAY SELECTIONコマンドを使用して、またはデザインモードのレコード表示ウィンドウ表示された、リストボックス、サブフォームとリストフォームです。
  - *リストボックス*: 少なくとも一つのセレクション型のリストボックスのが選択されていると、**サブレコード編集**に関連付けられたボタンが自動的に有効になります(標準アクションの使用を参照して下さい)。
  - *サブフォーム*: 親レコードのフォーム上に置かれます。サブフォームのレコードが選択されると、このボタンが使用可能になります。サブレコードをリストで選択し、**サブレコード編集**ボタンをクリックすると、そのサブレコードは (リスト中で直接編集、または割り当てられた詳細フォームを使用した) 編集モードに切り替わります (サブフォームのプロパティにより異なります)。
  - *リストフォーム*: このボタンはリスト上のレコードが選択されると使用可能になります。このボタンをクリックすると、そのレコードが編集モードに切り替わります。リストがMODIFY SELECTION / DISPLAY SELECTIONコマンドで表示されている場合、enterList引数の値に応じて、変更操作はリスト上または詳細ページ上で実行されます。デザインモードでは、詳細ページで変更が行われます(この動作はダブルクリックと同じです)。
- **サブレコード追加**: このアクションは3種類の状況下で利用できます。MODIFY SELECTION / DISPLAY SELECTIONコマンドを使用して、またはデザインモードのレコード表示ウィンドウ表示された、リストボックス、サブフォームとリストフォームです。
  - *リストボックス*: 少なくとも一つのセレクション型のリストボックスのが選択されていると、**サブレコード追加**に関連付けられたボタンが自動的に有効になります(標準アクションの使用を参照して下さい)。
  - *サブフォーム*: 親レコードのフォーム上に置かれます。サブフォームのレコードが選択されると、このボタンが使用可能になります。このボタンをクリックすると、4Dはその テーブルまたはリレートしたテーブルのレコードを新規に (リスト中に直接、または割り当てられた詳細フォームを開いて) 作成します (サブフォームのプロパティにより異なります)。
  - *リストフォーム*: このボタンは常に使用可能です。このボタンをクリックすると新しい空のレコードが作成されます。リストがMODIFY SELECTION / DISPLAY SELECTIONコマンドで表示されている場合、enterList引数の値に応じて、レコードはリスト上または詳細ページ上に追加されます。デザインモードではリストにレコードが追加されます。
- **サブレコード削除**: このアクションは2種類の状況下で利用できます。MODIFY SELECTION / DISPLAY SELECTIONコマンドを使用して、またはデザインモードのレコード表示ウィンドウ表示された、リストボックス、サブフォームとリストフォームです。
  - *リストボックス*: 少なくとも一つのセレクション型のリストボックスのが選択されていると、**サブレコード削除**に関連付けられたボタンが自動的に有効になります(標準アクションの使用を参照して下さい)。
  - *サブフォーム*: 親レコードのフォーム上に置かれます。サブフォームでサブレコードが選択されると、このボタンが使用可能になります。
  - *リストフォーム*: このボタンはリスト中で1つ以上のレコードが選択されると有効になります。このボタンをクリックするとダイアログが表示され、削除を実行するかどうか選択することができます。

  サブフォーム内でのボタンの利用に関してはを参照してください。
- **自動スプリッター**: この標準アクションを使用すると、フォーム上に独自のスプリッターを作成することができます。このアクションは透明ボタンにのみ指定することができます(ボタン 参照)。透明ボタンにこのアクションを指定すると、そのボタンはスプリッターと同じように機能します。例えばピクチャーをこの透明ボ タンにペーストすることにより、カスタマイズされたスプリッターインターフェースを作成できます。スプリッターに関する詳細はスプリッターを参照してください。
- **取り消し**: 前回実行した動作をキャンセルします (**編集**メニューの**取り消し**コマンドと同じ)。**取り消し**と**キャンセル**を混同しないでください。キャンセルはレコードに対して行われた変更をすべて取り消し、出 力フォームへ戻ります。
- **やり直し**: 前回キャンセルした動作をやり直します (**編集**メニューの**やり直し**コマンドと同じ)。
- **カット**: 選択部分を削除し、それをクリップボードに配置します。
- **コピー**: 選択部分のコピーをクリップボードに配置します。
- **ペースト**: 挿入ポイントがある場所にクリップボードの内容を挿入します。
- **クリア**: 選択部分を削除します。何も選択されていない場合は、カーソルが置かれているエリア全体を消去します(入力可エリアのみ)。
- **すべてを選択**: その時点で選択可能な項目をすべて選択します。
- **クリップボード表示**: 新しいウインドウを開き、そこにクリップボードの現在の内容を表示します。
- **データベース設定**: 標準のデータベース設定ダイアログを表示します (注参照)。
- **アプリケーション**: 4Dのアプリケーションモードのウィンドウとメニューバーを最前面にします。
- **終了**: "よろしいですか?"という確認ダイアログボックスを表示し、これが確定されると4Dアプリケーションを終了します。確定されない場合この操作がキャンセ ルされます。
  オブジェクトメソッドが関連付けられたボタンに対してこのアクションを指定すると、次の順に処理が行われます。まず初めに、確認ダイアログボックスが表示されま す。これが確定されると4Dはオブジェクトメソッドを実行します。実行後アプリ ケーションが終了します。
  **Mac OSにおける注意**: Mac OS Xでは、カスタムメニューコマンドに**環境設定**や**終了**アクションを関連付けた場合、データベースがアプリケーションモードで動作する時には、それらのコマンドが自動的にアプリケーションのシステムメニューに配置されます。このメカニズムにより、Mac OS Xにおける**終了**コマンドを管理しやすくなります。
- **デザインモードに戻る**: 4Dデザインモードのウィンドウやメニューバーを最前面にします。
  デー タベースがインタープリターモードで動作している場合、このアクションはデザインモードのカレントウィンドウを表示します。データベースがコンパイルモー ドで動作している場合、このアクションはカレントテーブルのレコードウィンドウを表示します (コンパイルモードではレコードへのアクセスのみ可能です)。
- **MSC**: Maintenance & Security Centerウィンドウを表示します。
- **ページ移動**: タブコントロールでのみ利用可能です。このアクションが有効になると、4Dは自動で選択されたタブに対応するページを表示します。詳細はページ指定アクションを参照してください。

以下のアクションはWebエリアを含むフォームで利用できます (Webエリア参照)。

- **前のURLを開く**: このアクションはWebエリア中でユーザーが実行した一連のブラウジングの流れで、前のURIを開きます。前のURLがない場合、つまりユーザーがエリア でひとつのページしか表示していない場合、このアクションが割り当てられたボタンやメニューコマンドは無効となります。
- **次のURLを開く**: このアクションはWebエリア中でユーザーが実行した一連のブラウジングの流れで、次のURIを開きます。次のURLがない場合、つまりユーザーがページを前に戻ったことが無い場合、このアクションが割り当てられたボタンやメニューコマンドは無効となります。
- **現在のURLを再読み込み**: このアクションはWebエリアの現在の内容をリロードします。
- **URLのロードを停止**: このアクションはWebエリアへの現在のURLのページやオブジェクトのロードを停止します。

## ミラー効果 (Windows)

(プロパティリストの"ピクチャー"テーマにある) **ミラー効果 (Windows)** オプションは、ピクチャーを使用するアクティブオブジェクトに適用できます。

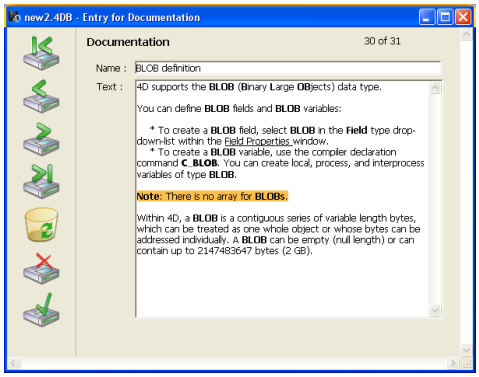
Windowsでこのオプションを使用すると、"right-to-left"モードに設定されたアプリケーションで、オブジェクトに割り当てられたピクチャーオブジェクトが表示される際、反転されます。このオプションはピクチャーフィールドや変数、ピクチャーボタン、ピクチャーポップアップメニュー、3Dボタン、3Dチェックボックス、3Dラジオボタン、ピクチャーラジオボタン、さらにスタティックピクチャーに設定できます。

このオプションについてはスタティックピクチャーを使用するで説明しています。

# マルチスタイル (リッチテキストエリア)

## 概要

4Dでは文字ごとににスタイルを設定できるリッチテキストエリアを使用できます。例えばテキストに太字、イタリック、文字色を持たせることができます:




この新しい機能は文字およびテキストタイプのフィールドや変数およびリストボックスのセルに適用されます。ページおよびリストフォームで、表示および印刷がサポートされます。

**注:** 以下のコンテキストではリッチテキストエリアを使用できません: 入力フィルター、クイックレポート、ラベルエディター

このリッチテキスト機能を設定するためにプロパティリストのオプションを使用します。

利用可能な属性はフォント、サイズ、スタイル、文字色、そして背景色です。リッチテキストエリアでスタイルを更新するには 2つの方法があります:

- 実行中に自動ポップアップメニューを使用する (このメニューを利用可能にするにはプロパティリストで設定します)。
- ST SET ATTRIBUTESコマンドを使用する。

リッチテキストエリアにて、スタイル属性は <SPAN> タグとして格納されます。テキストエリアが表示されると、これらのタグを4Dが解釈します。これは開発者がプログラムを使用してスタイル属性を指定したり 変更したりできることを意味します。4Dがサポートする <SPAN> タグは後述の通りです。
[#cmd id="1092"/]コマンドはスタイルタグなしの生テキストを取り出します。

リッチテキストエリアを管理するコマンドとサポートされているタグの詳細に関しては、4D *Language Reference* マニュアルのStyled Text の章を参照して下さい。

## リッチテキスト管理プロパティ

リッチテキストのサポートを有効にして設定するためのプロパティは文字またはテキスト型の入力可変数、フィールド、リストボックスセルで利用できます。

### マルチスタイル

このオプション ("テキスト" テーマ ) は、選択されたエリアで指定されたスタイルを利用可能にするかどうかを指定するものです。このオプションがチェックされていると、4D はエリア中の <SPAN> HTMLタグをスタイル属性として解釈します。

デフォルトでこのオプションはチェックされていません。

### デフォルトスタイルタグを格納

このオプションは**マルチスタイルオプション**がチェックされているときにのみ"テキスト"テーマに表示されます。
このオプションがチェックされていると、エリアはたとえスタイルの変更が行われていなくても、テキストとともにスタイルタグを格納します。この場合、タグはデフォルトスタイルに対応します。このオプションがチェックされていないと、更新されたスタイルタグのみが格納されます。

以下のようにスタイルが変更されたテキストがあります:

What a beautiful day

"デフォルトスタイルタグを格納"オプションがチェックされていない場合、エリアは更新されたスタイルのみを格納します。格納される内容は以下のようになります:

```
What a<SPAN STYLE="font-size:13.5pt">beautiful</SPAN> day !
```

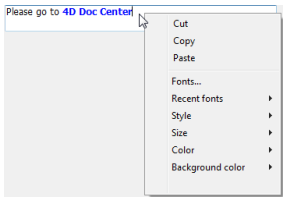
オプションがチェックされていると、エリアはすべてのフォーマット情報を格納します。先頭の汎用タグはデフォルトスタイルを定義し、変更されたスタイルはネストされたタグに書き込まれます。格納される内容は以下のようになります:

```
<SPAN  STYLE="font-family:'Arial';font-size:9pt;text-align:left;font-weight:normal;font-style:normal;text-decoration:none;color:#000000;background-color:#FFFFFF">What  a <SPAN STYLE="font-size:13.5pt">beautiful</SPAN>  day!</SPAN>
```

### コンテキストメニュー

このオプション ("入力"テーマ ) は**マルチスタイル**オプションが チェックされているときにのみ表示されます。

**コンテキストメニューオプション**が選択されていると、ユーザがエリアに入力中、右クリックでポップアップメニューが表示されるようになります:




このポップメニューでは、以下の様なコマンドを使用することができます:

- 標準のテキスト編集コマンド(切り取り、コピー、貼り付け)
- **フォント...**: フォントシステムダイアログボックスを表示させます。
- **最近使用したフォント**: セッション中に最近使用されたフォント名を表示します。リストには最大で10フォントまで表示されます(それ以上は古いものから置き換えられていきます)。デフォルトではリストは空になっているのでこのオプションは表示されていません。このリストは SET RECENT FONTS と FONT LIST コマンドを使用して管理することができます。
- スタイルの変更を行うためのコマンド:スタイル、サイズ、カラー、背景色

ユーザーがこのポップアップメニューを使用してスタイル属性を編集すると、4DはOn After Editフォームイベントを生成します。

**注:**

- ST SET ATTRIBUTES コマンドを使用してスタイ ルを編集することもできます。この場合フォームイベントは生成されないことに留意してください。
- Mac OSでは"strikethrough (取り消し線)"スタイルをサポートしていません。しかし対応するタグをプログラムで使用することはできます。

## リッチテキストの処理

### コピー/ペーストとドラッグ&ドロップ

サポートされるスタイル属性 (フォント、サイズ、スタイルそしてカラー) は以下のケースでドラッグ&ドロップやコピー/ペースト時にも保持され ます:

- 4D内での異なるリッチテキストエリア間 (テキスト変数/フィールド、そしてリストボックス)
- 4D Writeエリアと4Dリッチテキストエリア間
- 外部スタイル付きテキストエリアと4Dリッチテキストエリア間

その他の場合、スタイルはコンテキストに基づいて保持されます。

### マウスオーバー

マルチスタイルエリア上をマウスオーバーしたときの機能に、二つ新しいものが追加されました:

- テキストセレクションの上を移動すると、カーソルは矢印に変化します:
  What a beautiful day What a beautiful day
- URL上を移動すると、そのアドレスがtipの中に表示されます:

**注：** URLのリンクは ST INSERT URL コマンドを使用することで挿入できます。

## 複数行

"入力"テーマの特定のプロパティは、フォーム上のテキストエリア (フィールドや変数) が表示や印刷される際の設定に関連します:

- 一行のエリアの行の終わりの単語の表示に関する設定
- テキストエリア中での自動改行設定

**注:** 文字タイプのオブジェクトでは改行は使用できません。

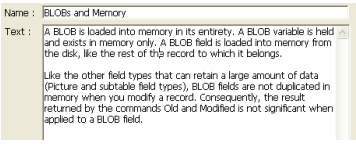
### 複数行

このプロパティは文字およびテキストタイプの変数やフィールド、また入力可や入力不可両方で使用できます。**はい**、**いいえ**、**自動**が選択できます。

#### 複数行 = 自動

一行のエリアで、行の最後にある単語はエリアからはみ出し、改行は行われません。

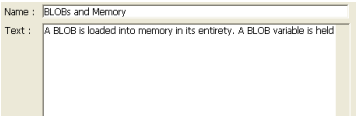
複数行のエリアでは、4Dは自動で改行します:




#### 複数行 = いいえ

一行の場合、行の最後の単語はエリアからはみ出し、改行は行われません。

改行は行われません。テキストは常に一行で表示されます。文字やテキストのフィールドや変数が改行文字を含んでいる場合、エリアが更新されるとすぐに最初のキャリッジリターンより後のテキストが取り除かれます:




#### 複数行 = はい

この値を選択すると、追加のオプション**ワードラップ**が表示されます。このオプションの値も設定する必要があります。

### ワードラップ

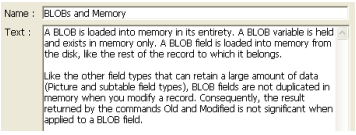
このプロパティは**複数行**プロパティが**はい**に設定されている場合にのみ表示されます。3つの選択肢**はい**、**いいえ**、**自動**があります。

**注:** このプロパティが正しく動作するためには、テキストオブジェクトがスクロールバーを持っていてはなりません。

#### ワードラップ= 自動

一行のエリアの場合、テキストは最初の改行まで、あるいは単語全体が表示できる最後の単語までが表示されます。4Dは改行を挿入します。下矢印キーを押すことでエリアの内容をスクロールできます。

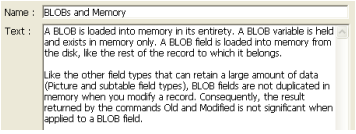
複数行の場合、4Dは自動改行を実行します:




#### ワードラップ = はい

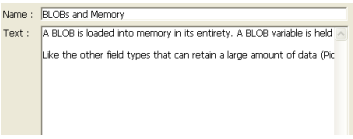
一行エリアの場合、全体が表示できる最後の単語までが表示されます。4Dは改行を挿入します。下矢印キーを押すことで、エリアの内容をスクロールできます。

複数行のエリアの場合、4Dは自動改行を実行します:




#### ワードラップ = いいえ

4Dはいっさい自動改行を行いません。表示可能な最後の単語はエリアをはみ出します。テキストタイプのエリアでは改行がサポートされます:



## スクロールバー

あるタイプのオブジェクト (テキストやピクチャー型のフィールドおよび変数オブジェクト、階層リスト、リストボックス、サブフォーム) にはスクロールバーを割り当てることができます。プロパティリストやOBJECT SET SCROLLBARコマンドを使用したプログラミングでこれらのプロパティを管理できます。

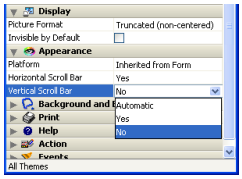
スクロールバーを割り当てるには、プロパティリストの"アピアランス"テーマにある**水平スクロールバー**や**垂直スクロールバー**プロパティを使用します。テキスト型のオブジェクト、階層リスト、リストボックス、そしてサブフォームでは、これらのプロパティをチェックボックスを使用して設定します。

**注:** テキストフィールドや入力可オブジェクトにスクロールバーがない場合、矢印キーを使用してスクロルできます。

### ピクチャーのスクロールバー

ピクチャー型のオブジェクトは、表示フォーマットが"トランケート (中央合わせなし)"に設定されているときに、スクロールバーを持つことができます。ピクチャー表示フォーマットに関する詳細はGET LIST ITEM PARAMETERの"ピクチャーフォーマット"を参照してください。

この場合、**水平スクロールバー**と**垂直スクロールバー**が、ピクチャーオブジェクトのプロパティリスト内で有効になります。スクロールバーを持つことのできる他のオブジェクトと異なり、ピクチャー型のオブジェクトではメニューを使用して3つのオプションからスクロールバーを設定します:




- **はい**: スクロールバーは必要のない場合 (ピクチャーがフレームより小さい場合) でも常に表示されます。
- **いいえ**: スクロールバーは表示されません。
- **自動**: スクロールバーは必要なとき (ピクチャーがフレームより大きい場合) に表示されます。

#### スクロールキー

4D v13.2より、スクロールバーが有効なピクチャーエリアでスクロールキーを利用できるようになりました。以下のキーがサポートされています:

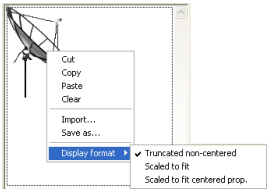
| キー | アクション |
|---|---|
| 矢印キー | 矢印方向にスクロールが行われます (スクロールバーカーソルをクリックするのと同様) |
| *Alt* + Arrow keys | 矢印方向にピクセル毎にスクロール |
| Page Up / Page Down | ページ毎の縦スクロール (エリアの高さに相対) |
| *Shift* + Page Up / Page Down | ページ毎の横スクロール (エリア幅に相対) |
| Home | ピクチャーの最上端を表示 |
| End | ピクチャーの最下端を表示 |
| *Shift* + Home | ピクチャーの左端を表示 |
| *Shift* + End | ピクチャーの右端を表示 |
| マウスホイール | 縦スクロール (4D v12で利用可能) |
| *Shift* + マウスホイール | 横スクロール (4D v12で利用可能) |

4D v13.2より、ピクチャーのスクロールにより On Picture Scroll Form eventが生成されます。

## コンテキストメニュー (ピクチャー)

ピクチャー型のフィールドや変数に、自動コンテキストメニューを割り当てることができます。これを行うには、プロパティリストの"入力"エリアで、**コンテキストメニュー**オプションにチェックをいれます。

このオプションにチェックが入れられると、フォームがデザインモードやアプリケーションモードで実行されたとき、エリアでコンテキストメニューが使用できるようになります。ピクチャー上で右マウスボタンをクリックすると、編集や表示コマンドにアクセスできます:




標準の編集コマンド (**カット**、**コピー**、**ペースト**そして**クリア**) に加え、ファイルからピクチャーを読み込むために使用することのできる**読み込み...**コマンド、ピクチャーをディスクに保存する為に使用する**別名で保存...**コマンドなどがあります。これら2つのコマンドはネイティブなピクチャー管理を使用します。4Dがサポートするネイティブフォーマットのピクチャーの保存や読み込みに使用できます。またメニューを使用してピクチャーの表示フォーマットを**トランケート (中央合わせなし)**、**スケーリング**そして**スケーリング (中央合わせ/プロポーショナル)** に変更できます。このメニューを使用した表示フォーマットの変更は一時的なものであり、レコードには保存されません。ピクチャー表示フォーマットについてはGET LIST ITEM PARAMETERを参照してください。

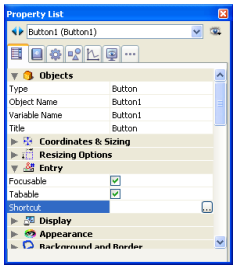
ピクチャーフィールドや変数が入力不可の場合、**コピー**、**別名で保存...**そして表示フォーマットコマンドのみを利用できます。

**注:** テキストフィールドや変数にもコンテキストメニューを設定できます。この場合は**マルチスタイル**プロパティによります (マルチスタイル (リッチテキストエリア)参照)。

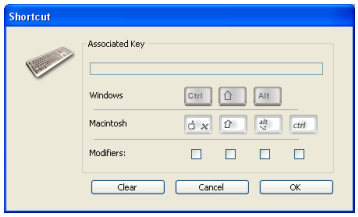
## キーボードショートカット

ボタンとチェックボックスにキーボードショートカットを割り当てることができます。結果ユーザーはマウスを使用する代わりにキーボードからボタンアクションを実行したり、チェックボックスを選択したりできます。

ショートカットの設定を行うには、プロパティリストの"入力"テーマ内、ショートカットプロパティの[...]ボタンをクリックします:




以下のダイアログが表示されます:




キーボードショートカットをタイプします。例えば**Ctrl+H**を使用したい場合、**Ctrl**キーを押しながら**H**を押します。すると割り当てキーエリアに文字Hが表示され、下部のCtrlキーのチェックボックスがチェックされます。モディファイアキーを使用する必要はありません。どのようなキーでも単独でショート過渡として使用することができますが、ほとんどの場合これは推奨されません。必要であれば手作業でモディファイアキーを追加したり取り除いたりできます。

元に戻すには**クリア**をクリックします。設定を終了するには**OK**をクリックします。

プロパティリストにはオブジェクトに割り当てられたキーボードショートカットが表示されます。ショートカットを再設定するには、ショートカットダイアログを開き、使用したいキーコンビネーションを押します。

**注:** カスタムメニューコマンドにもショートカットを割り当てることができます。2つのショートカットに衝突があると、アクティブオブジェクトが優先されます。メニューへのショートカットの割り当てについてはメニュープロパティを設定するを参照してください。

## ウィンドウの配置の記憶

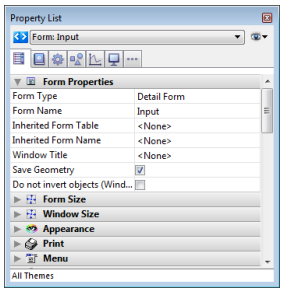
4D には、ウィンドウ独自の表示方法("配置")を、ウィンドウを閉じる際に自動的に記憶するという機能があります。これにより、ユーザーは作業を終了したときと同じ作業環境にいつでも戻って来られるようになります。

この機能にて記録されるのは、ウィンドウの座標とそのウィンドウ内に含まれるオブジェクトの位置、それからいくつかのオブジェクトのカレントの状態、すなわちタブコントロールでどれが選択されているか、などです。

**注:** この新機能は、フォームが閉じたときと同じサイズで再度開かれたときのみ有効です。結果的に、この機能は主にOpen form windowコマンドに * 演算子を渡してウィンドウが開かれることを想定しています。

**「配置を記憶」のオプション**

自動配置記憶を有効化するためには、フォームプロパティ内にある**配置の記憶**オプションをチェックする必要があります:




このオプションがチェックされているとき、セッション中にどのように変更したかに関係なく、以下の複数のフォームパラメーターが 4D によって自動的に保存されます:

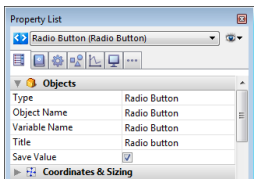
- カレントのページ
- それぞれのフォームの配置、大きさ、表示状態(リストボックス列のサイズと表示状態も含みます)。

**注:** このオプションは、OBJECT DUPLICATEコマンドを使用して作成されたオブジェクトに対しては無効です。このコマンドを使用しても使用環境を復元させるためには、デベロッパがオブジェクトの作成・定義・配置の手順をその都度行わなければなりません。

このオプションが選択されているとき、一部のオブジェクトに置いては**値を記憶**のオプションが選択可能になります(詳細は次の項目を参照して下さい)。

**「値を記憶」のオプション**

新しい**値を記憶**のオプションはプロパティリストの"オブジェクト"のテーマ内にあります:




このオプションは以下の場合に限り有効です:

- フォームの「配置を記憶」のオプションのオプションがチェックされている
- 選択されたオブジェクトがフォームの全体的な配置に関係するオブジェクトであること。
  例えばチェックボックスは、ウィンドウ内にて追加のエリアを表示たり隠したりするのにこの値を使用することができるので、チェックボックスにはこのオプションが存在します(以下の表を参照して下さい)。

「値」を保存できるオブジェクトと保存できる値は、以下の表の通りです:

| オブジェクト | 保存される値 |
|---|---|
| ラジオボタン | 関連付けられた変数の値(1、0、ボタンにおいてはTrueまたはFalse など。型による) |
| 3D ラジオボタン | 関連付けられた変数の値(0, 1) |
| チェックボックス | 関連付けられた変数の値(0, 1, 2) |
| 3D チェックボックス | 関連付けられた変数の値(0, 1) |
| タブ | 選択されているタブの番号 |
| ポップアップ/ドロップダウンリスト | 選択されている項目の番号 |
| ピクチャーポップアップメニュー | 選択されている項目の番号 |

**情報の保存と使用に関して**

4D では **Open form windowコマンド(フォーム名 :*)**を使用して作成されたウィンドウを閉じた際にそのウィンドウの座標を保存します。Windows 環境下では最大化した状態も保存されます。

これらの情報と、任意で保存できる情報(**配置**や**値**など)は、ウィンドウが閉じられたときにマシンのカレントのユーザーフォルダに*jsonフォーマット*で保存されます。これにより、たとえ"デフォルトユーザー"アカウントを使用したときでも個別のマシンを使って接続するそれぞれのユーザーごとに、その環境を保存することが出来ます。

この情報は、フォームが閉じられたときと同じ寸法にて再度開かれたときに使用され、閉じられたときに再度保存されます。この原理には、Open form window(*)コマンドが使用されたか、デベロッパが座標を保存するための独自のシステムを設定されていることを前提としています。

保存された情報は以下の順序で復元され、適用されます:

- ウィンドウのサイズとポジションはOpen form windowコマンドが実行されたときに復元されます。
- 変数の値はフォームが On Load イベントを呼び出す前にロードされたときに復元されます。
- カレントのページは On Load イベントを呼び出す前に復元されます。
- それぞれのオブジェクトの位置、サイズ、そして表示状態は On Load イベントのすぐ後に復元されます。

サブフォームオブジェクトのプロパティは、同じ順番に従って自動的に保存・適用されます。

***警告:** "配置を記憶"、"値を記憶"を使用して保存された情報は、フォームのオブジェクトがデザインモードで変更(リサイズ、移動、追加、削除または改名)されるたび元の状態へとリセットされてしまいます。そのため、この機能はユーザー設定などの恒久的な値を保存するためには**使用しない**ことを強く推奨します。*

### Examples

**スプリッターの記憶**

フォームのスプリッターの相対位置を保存したい場合を考えます。この場合、**配置を保存**のオプションをチェックするだけです。
フォームが開いたとき、以下の様になるはずです:

ユーザーがウィンドウをリサイズしたりスプリッターを動かしたりしたとします。オブジェクトはそれぞれのプロパティに応じてリサイズされます。ユーザーがウィンドウを閉じた後、再びフォームが開かたとき、オブジェクトは最新の状態を復元します:

**折り畳みエリアの記憶**

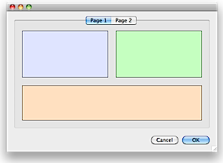
フォーム内において、3Dチェックボックスによって管理される、一つ以上の折り畳みエリアがある場合、そのチェックボックスはエリアが折りたたまれている場合は右向きの三角形を表示し、エリアが展開されている場合には下向きの三角形を表示します。こういったエリアを設定するにはいくつかの方法(オブジェクトの移動または表示状態の変更、異なるフォームページの使用等)がありますが、どのケースにおいても、ウィンドウのサイズが変更になることがあります。

異なるセッション間で折り畳みエリアの状態を保存したい場合、以下の様にしてください:

- **フォームの配置**を保存のオプションにチェックをし、オブジェクトのカレントページ、位置、表示状態が保存されるようにしてください。
- 3Dチェックボックスの**値を保存**のオプションにチェックをし、関連付けられた変数の値(展開か折り畳みを表す0または1)が保存されるようにしてください。

**タブの記憶**

フォーム内にて、"Goto Page" の標準アクションを持ったタブを設定したとします:




この場合、配置を正しく保存させるためには、フォーム内の**配置を保存**にチェックをし、タブオブジェクトの**値を保存**のオプションをチェックします:

# リストボックス


概要

リストボックス特有のプロパティ

リストボックス列特有のプロパティ

リストボックスヘッダー特有のプロパティ

リストボックスフッター特有のプロパティ

階層リストボックスの作成と利用

標準アクションの使用

リストボックスにフィールドを表示する

リストボックスにSQLクエリの結果を表示する

## 概要

リストボックスは複合アクティブオブジェクトであり、同期化された列としてデータの表示や入力を行えるエリアです。リストボックスには"グループ化したスクロールエリア"と同じ基本的な機能があり、また新たに拡張された機能も備えています(値入力、列の並び替え、外観のカスタマイズ、列の移動など)。リストボックスタイプのオブジェクトはすべて4Dのフォームエディターで設定可能であり、またプログラムから処理することもできます。

| Last name | First name | Position |
|---|---|---|
| Column4{1} | Column5{1} | Column3{1} |
| Column4{2} | Column5{2} | Column3{2} |
| Column4{3} | Column5{3} | Column3{3} |
| Column4{4} | Column5{4} | Column3{4} |
| Column4{5} | Column5{5} | Column3{5} |
| Column4{6} | Column5{6} | Column3{6} |
| Column4{7} | Column5{7} | Column3{7} |
| Column4{8} | Column5{8} | Column3{8} |
| Column4{9} | Column5{9} | Column3{9} |
| Column4{10} | Column5{10} | Column3{10} |
| Column4{11} | Column5{11} | Column3{11} |
| Column4{12} | Column5{12} | Column3{12} |
| Column4{13} | Column5{13} | Column3{13} |
| Column4{14} | Column5{14} | Column3{14} |
| Column4{15} | Column5{15} | Column3{15} |

この章ではフォームエディターでリストボックスタイプのオブジェクトを作成し、設定するための原則について説明します。プログラムでこれらオブジェクトを管理する際の情報については、4Dランゲージリファレンスマニュアルのリストボックスを参照してください。

### セレクションタイプと配列タイプ

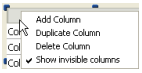
リストボックスには2つのタイプ、**セレクションタイプリストボックス**と**配列タイプリストボックス**があります。

- **セレクションタイプリストボックス**: 行数はカレントセレクションまたは命名セレクションに含まれるレコード数に基づき決定されます。各列にはフィールド (例えば[Employees]LastName) または4D式を割り当てます。式は1つ以上のフィールド (例えば[Employees]FirstName+" "[Employees]LastName) または単にフォーミュラ (例えばString(Milliseconds))を使用できます。式にはプロジェクトメソッド、変数、あるいは配列項目を指定できます。
  カレントセレクションに基づくリストボックスの場合、データベース側で行われた更新は自動でリストボックスにも反映されます。またその逆も同様です。つまりカレントセレクションの場合、常に両方で同じとなります。
- **配列タイプリストボックス**: 行数は配列の要素数により決定されます。リストボックスのそれぞれの列には4D配列を割り当てます。デフォルトで4Dは各列に"ColumnX"という名 前の配列変数を割り当てます。列のプロパティで配列変数名を変更できます。このタイプのリストボックスでは、入力あるいは表示される値は4Dランゲージで制御します。列に選択リストを割り当てて、データ入力を制御することもできます (リストボックス列特有のプロパティ参照)。

フィールドを割り当てた列と配列を割り当てた列を同時にひとつのリストボックスで使用することはできません。

### 主な特徴

リストボックスには1つ以上の列があり、その内容が自動的に同期化されます。デフォルトでリストボックス作成時は1つの列が含まれています。コンテキストメニュー (任意の列、または列ヘッダをクリック) やリストボックスのプロパティで列数を変更することができます (列の追加、複製、削除)。




理論上、列数に制限はありません(マシン資源に依存します)。

リストボックスは**オブジェクト全体**、**列**、**列ヘッダー**、**列フッター**という異なる4つの部分から成り立ちます。フォームエディターでは、これらの各部分を別々に選択することもできます。各部分には独自のオブジェクトと変数名、ならびに固有のプロパティがあります。例えば列の数やそれぞれの行の代替カラーは、リストボックスオブジェクトのプロパティで定義します。各列の幅は列のプロパティで指定し、ヘッダのフォントはヘッダーのプロパティで定義します。

**注:** セレクションタイプのリストボックスでは、列に変数名はありません。

配列タイプのリストボックスは**標準モード**または**階層モード**で表示できます。階層モードで表示されるリストボックスは以下に説明するような特別なメカニズムを使用します。

### リストボックスを使用する

実行中、リストボックスはリストとしてデータを表示し、入力を受け付けます。
セルを編集可能にするには (列が入力を受け付けるようにするには)、セル上で2回クリックを行います:

| Last name | First name |
|---|---|
| James | Henry |
| Jameson | Marc |

**注**: 詳細は4Dランゲージリファレンスマニュアルの入力の管理を参照してください。

ひとつのリストボックスセル内に複数行のテキストを入力し、表示できます。改行を追加するには:

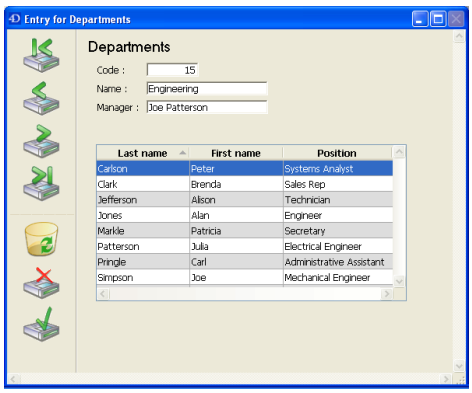
- Windows: **Ctrl+Carriage** を押します。
- Mac OS: **Option+Carriage** を押します。

行の高さは自動では調整されないことに留意してください。

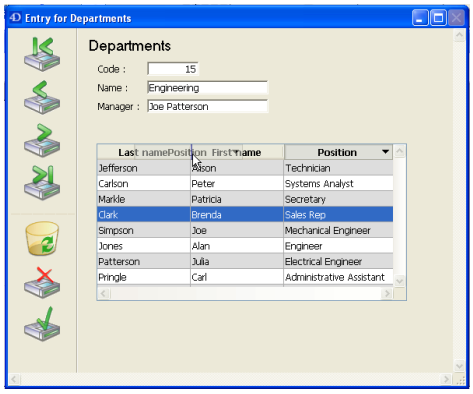
ヘッダーをクリックすると、列の値をソートできます (標準ソート)。並び替えは文字順で行われ、クリックするごとに昇順と降順が入れ替わります。すべての列が自動で同期されます。

**注**: 詳細は4Dランゲージリファレンスマニュアルのソートの管理を参照してください。

またそれぞれの列幅を変更することもできます:




ユーザーはマウスを使用して列や行の順番を入れ替えることもできます (そのアクションが許可されていれば):




**注:** セレクションタイプのリストボックスでは、マウスによる行の入れ替えは行えません。

最後に、ユーザーは標準のショートカットを使用して1つ以上の行を選択できます。**Shift+クリック**で連続した行を、**Ctrl+クリック** (Windows) や **Command+クリック** (Mac OS) で非連続行を選択できます。

これらすべての特徴は、リストボックス、列、ヘッダー、フッターそれぞれのプロパティを使用して制御できます。この点については後述します。

**注:** 階層モードのリストボックスの特徴については階層リストボックスの作成と利用で説明します。

#### リストボックスの印刷

リストボックスは"プレビュー"モード (リストボックスがフォームに表示されている状態のピクチャー)、または"詳細"モード (内容に基づき可変サイズ) で印刷できます。詳細は4Dランゲージリファレンスのリストボックスの印刷を参照してください。

#### スタイルとカラーの管理

リストボックスの背景色、フォントカラー、そしてフォントスタイルを設定するためにはいくつかの方法があります。リストボックスオブジェクトまたは列のプロパティリストの使用、リストボックスまたは列ごとの配列またはメソッドの使用、そしてセルレベルでの定義(マルチスタイルテキスト時)です。優先順位と継承の原理はそのままです。

##### 優先順位

同じプロパティに異なる値が複数のレベルにわたって適用された場合、以下の優先順位が適用されます:

*優先度高* セル単位(マルチスタイル使用時)
列の配列/メソッド
リストボックスの配列/メソッド
列のプロパティ
*優先度低* リストボックスのプロパティ

例として、リストボックスのプロパティにてリストボックスにフォントスタイルを設定し、列のスタイル配列を使用して列に対して異なるスタイルを設定した場合、後者の方が有効となります。

以下の様な、グレー/淡いグレーを交互に繰り返す行の背景色がリストボックスのプロパティで定義されたリストボックスについて考えます。同時に、行の中の少なくともどれか一つの値が負の値である行に関しては背景色がオレンジ色になるような背景色配列が設定されていたとします:

```
<>_BgndColors{$i}:=0x00FFD0B0 // オレンジ
<>_BgndColors{$i}:=-255 // デフォルト値
```

| Header1 | Header2 | Header3 |
|---|---|---|
| 21483 | 9031 | 27290 |
| 24151 | 21990 | -923 |
| 21351 | 2982 | 18009 |
| 8089 | 12898 | 20941 |
| 13001 | -802 | 22059 |
| 4321 | 16826 | 11303 |
| 24082 | 26214 | 22380 |
| 16680 | 23651 | 20403 |
| -2678 | 24818 | 29896 |
| 25639 | 2691 | 9687 |
| 28794 | 26941 | 21486 |
| 26083 | 21092 | 13476 |
| 27928 | 4092 | 15441 |
| 19987 | 28211 | 21191 |
| 7996 | 6300 | 4089 |
| -1063 | 13388 | 23683 |
| 13008 | 7470 | 19897 |
| 5388 | 26918 | 13547 |
| 28559 | 27007 | 8365 |
| 28454 | 22646 | 13824 |

次に、負の値が入っているセルを濃いオレンジの背景色にしたい場合を考えます。これをするためには、それぞれの行に対して背景色を適用します。例えば、*<>_BgndColor_1*, *<>_BgndColor_2* そして*<>_BgndColor_3* のようにです。これらの配列の値はリストボックスのプロパティや一般的な背景色の設定より優先されます:

```
<>_BgndColorsCol_3{2}:=0x00FF8000 // 濃いオレンジ
<>_BgndColorsCol_2{5}:=0x00FF8000
<>_BgndColorsCol_1{9}:=0x00FF8000
<>_BgndColorsCol_1{16}:=0x00FF8000
```

| Header1 | Header2 | Header3 |
|---|---|---|
| 21483 | 9031 | 27290 |
| 24151 | 21990 | -923 |
| 21351 | 2982 | 18009 |
| 8089 | 12898 | 20941 |
| 13001 | -802 | 22059 |
| 4321 | 16826 | 11303 |
| 24082 | 26214 | 22380 |
| 16680 | 23651 | 20403 |
| -2678 | 24818 | 29896 |
| 25639 | 2691 | 9687 |
| 28794 | 26941 | 21486 |
| 26083 | 21092 | 13476 |
| 27928 | 4092 | 15441 |
| 19987 | 28211 | 21191 |
| 7996 | 6300 | 4089 |
| -1063 | 13388 | 23683 |
| 13008 | 7470 | 19897 |
| 5388 | 26918 | 13547 |
| 28559 | 27007 | 8365 |
| 28454 | 22646 | 13824 |

新しい LISTBOX SET ROW FONT STYLE コマンドと LISTBOX SET ROW COLOR コマンドを使用しても同じ結果を得ることができます。こちらを使った方がコマンドが動的に配列を作成するので、列ごとのスタイル/カラー配列を事前に設定するのをスキップすることが出来るという利点があります。

##### 継承

それぞれの属性(スタイル、カラー、背景色)について、デフォルトの値を使用した場合、属性の継承が行われるようになっています:

- セル属性について : 行の値を受け継ぎます
- 行属性について : 列の値を受け継ぎます
- 列属性について : リストボックスの値を受け継ぎます

このように、オブジェクトに高次のレベルの属性の値を継承させたい場合は、定義するコマンドの中に -255(デフォルト値) を渡すか、対応するスタイル/カラー配列の要素の中に直接渡して下さい。

以下の様な、標準のフォントスタイルで行の背景色が交互に変わるリストボックスを考えます:

| | | | | | |
|---|---|---|---|---|---|
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | standard | standard |

以下の様な変更を加えます:

- リストボックスオブジェクトの**行背景色配列**プロパティを使用して、2行目の背景色を赤に変更します。
- リストボックスオブジェクトの**行スタイル配列**を使用して、4 行目のスタイルをイタリックに変更します。
- 5 列目の列オブジェクトの**行スタイル配列**を使用して、5 列目の二つの要素を太字に変更します。
- 1、2 列目の列オブジェクトの**行背景色配列**を使用して、1、2 列目の要素一つずつ、計二つの背景色を濃い青に変更します:

| | | | | | |
|---|---|---|---|---|---|
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | **standard** | standard |
| *standard* | *standard* | *standard* | *standard* | **standard** | *standard* |
| standard | standard | standard | standard | standard | standard |
| standard | standard | standard | standard | standard | standard |

リストボックスを元の状態に戻すには、以下の手順で元に戻せます:

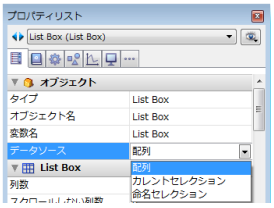
- 1、2 列目の背景色配列の要素 2 に定数 Listbox inherited 定数を渡します。これにより行の赤の背景色を継承します。
- 5 列目のスタイル配列の要素 3 と 4 に定数 Listbox inherited を渡します。これにより、要素 4 を除いて標準のスタイルを継承します(要素 4 はリストボックスのスタイル配列にて指定されたイタリックの属性を継承します)。
- リストボックスのスタイル配列の要素 4 に定数 Listbox inherited を渡します。これにより、4 行目のイタリックのスタイルが除去されます。
- リストボックスの背景色配列の要素 2 に定数 Listbox inherited を渡します。これにより元の、背景色が交互に変わるリストボックスの状態に戻すことが出来ます。

## リストボックス特有のプロパティ

フォームエディターでリストボックスオブジェクトを選択すると、プロパティリストには専用のテーマやプロパティが表示されます。リストボックス列(リストボックス列特有のプロパティ参照)や列ヘッダー (リストボックスヘッダー特有のプロパティ参照)や列フッター(リストボックスフッター特有のプロパティ参照)を選択すると追加の専用プロパティが利用できるようになります。

### オブジェクトテーマ

このテーマには**データソース**プロパティがあり、リストボックスのタイプを設定できます:




- **カレントセレクション**オプションを選択すると、指定したテーブルのカレントセレクションの各レコードごとに式、フィールド、メソッドが評価されます。
- **命名セレクション**オプションを選択すると、指定した命名セレクションに含まれる各レコードごとに式、フィールド、メソッドが評価されます。
- **配列**オプションを選択すると、リストボックスの各行に配列要素を表示することができます。
  SQLクエリーの結果をリストボックスに取得するためには、配列タイプを選択しなければなりません (リストボックスにSQLクエリの結果を表示する参照)。

### データソーステーマ

このテーマはセレクションタイプのリストボックスの場合にのみ表示されます。カレントセレクションの場合**マスターテーブル**プロパティが表示され、命名セレクションの場合**命名セレクションプロパティ**が表示されます。

- **マスターテーブル**: 使用するカレントセレクションが属するテーブルを指定します。このテーブルとそのテーブルのカレントセレクションが、リストボックスの列に割り当てられたフィールドの参照を形成します (フィールド参照やフィールドを含む式)。ある列が他のテーブルのフィールドを参照していてとしても、表示される行の数はマスターテブルのカレントレコード数となります。
  このプロパティに割り当てられるメニューには、データベースのすべてのテーブルが表示されます。フォームがどのテーブルに属しているか (テーブルフォームの場合) あるいは属していないか (プロジェクトフォーム) は関係ありません。デフォルトでデータベースの最初のテーブルがプロパティに表示されます。このプロパティの動作に関する詳細はリストボックスにフィールドを表示するを参照してください。
- **命名セレクション**: 使用する命名セレクションの名前を指定します。有効な命名セレクションの名前を入力しなければなりません。使用できるのはプロセスあるいはインタープロセス命名セレクションです。リストボックスの内容はこの命名セレクションに含まれるセレクションに基づきます。選択された命名セレクションは、リストボックスが表示される時点で存在し、有効でなければなりません。そうでない場合、リストボックスは空で表示されます。名前エリアを空にしても、リストボックスは空で表示されます。
  **注**: 命名セレクションはソート済みのレコードリストです。これはセレクション中のカレントレコードと並び順をメモリーに保持するために使用されます。詳細は4Dランゲージリファレンスの命名セレクションを参照してください。

### リストボックステーマ

このテーマには、基本的なオブジェクトプロパティがすべて集められています。

- **列数**: リストボックスに表示される列の数 (デフォルトは1)。コンテキストメニューを使用して (列または列ヘッダをクリック)、列数を変更することができます (列の追加、複製、削除):

  
- **スクロールしない列数**: ユーザーがリストボックスを横にスクロールしても、リストボックスの左側に常に表示される列の数を指定します。スクロールしない列は通常の列と同様、幅の リサイズ、セルへの入力などが可能です。リストボックス中での位置が固定されるだけであり、これは例えば第一列に行のタイトル (名前など) が表示されている場合に便利です。
  横スクロールしない列がプログラムで削除されると、横スクロールしない列数は1減少します。同様に横スクロールしないエリアに列がプログラムで挿入されると、この列は自動で横スクロールしない列になります。
- **ドラッグしない列数**: 実行時にドラッグで移動できない列の数。この値は、リストボックスの一番目の列から数えた固定列の数を表わします。すべての列の移動を禁止するに は、この値と列の合計数とを同じにしなくてはなりません。このプロパティは非表示の列も計算に入れる点に注意してください。
  このプロパティのデフォルト値は、階層モードが無効の (階層リストボックスオプションがチェックされていない) 場合0であり、これはすべての列が移動可能であることを意味します。階層モードが有効の場合 (配列タイプのリストボックスのみ)、このオプションは常に1以上です。
  **注**: これらの機能の違いについては後述の"横スクロールしない列とドラッグしない列"を参照してください。
- **列のヘッダを表示**: 列ヘッダの表示/非表示を指定します (デフォルトは表示)。
- **ハイライトセット**: このプロパティはセレクションタイプのリストボックスでのみ利用できます。このプロパティを使用して、リストボックス中でハイライトされたレコードを管理するために使用するセット名を指定します (**配列**データソースが指定されている場合、リストボックスに割り当てた変数と同じ名前のブール配列を使用します)。
  4Dは*ListBoxSetN* (N は 0 から始まり、フォーム上のリストボックスオブジェクトの数に従い増分されます) という名前のデフォルトセットを作成しますが、必要に応じてこの名前を変更できます。セットはローカル、プロセスおよびインタープロセスセットを使用できます (ネットワークトラフィックを制限するため、"$LBSet"のような名前のローカルセットの使用を推奨します)。指定されたセットは4Dが自動で管理します。ユーザーが1つ以上の行を選択すると、セットは即座に更新されます。プログラムを使用して行を選択したい場合、" セット"テーマのコマンドをこのセットに適用できます。
  **注**:
  - リストボックス行のハイライトステータスとテーブルレコードのハイライトステータスは完全に独立しています。
  - "ハイライトセット"プロパティに名前が指定されていない場合、リストボックス中で行を選択することはできません。
- **行スタイル配列** (配列タイプリストボックス) / **行スタイル** (セレクションタイプリストボックス): リストボックスの各行にカスタマイズされた文字スタイルを適用するために使用します。
  - 配 列タイプのリストボックスの場合、倍長整数配列名を入力します。この配列の各要素がリストボックスの行に対応します。この配列のサイズは 列に割り当てた配 列と同じでなければなりません。(メソッドを使用して) 配列を埋めるためには、"Font Styles"テーマの定数を使用します。定数を加算することでスタイルを合成できます。リストボックスのプロパティで設定したスタイルを行に適用するた めには、配列の対応する要素に値-255を渡します。
  - セレクションタイプのリストボックスの場合、式または変 数 (配列を除く) を入力します。式や変数は各行が表示される際に評価されます。式を指定する際にフォーミュラエディターを使用することができます。これを行うには、エリア を選択したときに表示される[...]ボタンをクリックします。"Font Styles"テーマの定数を使用することができます。
    以下の例は変数名を使用しています。**行スタイル**エリアに*CompanyStyle*と入力され、フォームメソッドにOn Display Detailで実行される以下のコードが書かれています:
    ```
    CompanyStyle:=Choose([Companies]ID;Bold;Plain;Italic;Underline)
    ```
- **行フォントカラー配列** (配列タイプリストボックス) / **行フォントカラー** (セレクションタイプリストボックス): リストボックスの各行にカスタマイズしたフォントカラーを適用するために使用します。RGBカラーを使用しなければなりません。この点に関する詳細は4D ランゲージリファレンスマニュアルのOBJECT SET RGB COLORSコマンドの説明を参照してください。
  - 配列タイプのリスト ボックスの場合、倍長整数配列名を入力します。この配列の各要素がリストボックスの行に対応します。この配列のサイズは 列に割り当てた配列と同じでなけれ ばなりません。"SET RGB COLORS"テーマの定数を使用できます。リストボックスプロパティで設定したフォントカラーを行に適用するには、配列の対応する要素に値-255を渡 します。
  - セ レクションタイプのリストボックスの場合、式または変 数 (配列を除く) を入力します。式や変数は各行が表示される際に評価されます。式を指定する際にフォーミュラエディターを使用することができます。これを行うには、エリア を選択したときに表示される[...]ボタンをクリックします。"SET RGB COLORS"テーマの定数を使用することができます。
    以下の例は変数名を使用しています。**行フォントカラー**エリアに*CompanyColor*と入力され、フォームメソッドにOn Display Detailで実行される以下のコードが書かれています:
    ```
    CompanyColor:=Choose([Companies]ID;Default background color;Default light shadow color;Default
    foreground color;Default dark shadow color)
    ```
- **行背景色配列** (配列タイプリストボックス) / **行背景色** (セレクションタイプリストボックス): リストボックスの各行にカスタマイズされた背景色を適用するために使用します。RGBカラー値を使用しなければなりません。この点に関する詳細は4Dラン ゲージリファレンスマニュアルのOBJECT SET RGB COLORSコマンドの説明を参照してください。
  - 配列タイプのリストボックスの場 合、倍長整数配列名を入力します。この配列の各要素がリストボックスの行に対応します。この配列のサイズは 列に割り当てた配列と同じでなければなりませ ん。"SET RGB COLORS"テーマの定数を使用できます。リストボックスプロパティで設定した背景色を行に適用するには、配列の対応する要素に値-255を渡します。
  - セレクションタイプのリストボックスの場合、式または変 数 (配列を除く) を入力します。式や変数は各行が表示される際に評価されます。式を指定する際にフォーミュラエディターを使用することができます。これを行うには、エリア を選択したときに表示される[...]ボタンをクリックします。"SET RGB COLORS"テーマの定数を使用することができます。
- **非表示行配列** (配列タイプリストボックスのみ): リストボックスの任意の行を非表示にするために使用します。ブール配列名を入力しなければなりません。この配列サイズはリストボックスの行数と同じでなけ ればなりません。配列の各要素はリストボックス行の表示/非表示ステータスを表します:
  - 行を隠すにはTrueを渡します。
  - 行を表示するにはFalseを渡します。

  配列が指定されていないか、配列が空の場合、リストボックスのすべての行が表示されます。デフォルトでつきあされた行は表示されます。
  行 の非表示はリストボックスの表示にのみ影響します。非表示行は依然として配列に存在し、プログラムで管理可能です。ランゲージコマンド、特にLISTBOX Get number of rowsやLISTBOX GET CELL POSITIONはリストボックス行の表示/非表示の影響を受けません。例えば10行あるリストボックスで先頭の9行が隠されている場合 、LISTBOX Get number of rowsは10を返します。
  ユーザーの視点から見た場合、リストボックス中に非表示行があることは視認できません。表示されている行だけが選択できます (例えばすべてを選択コマンドを使用して)。
- **選択モード**: リストボックス行の選択モードを指定するために使用します。3つのモードから選択できます:
  - **なし**: 行を選択することはできず、データを入力することもできません。選択とデータ入力はプログラムからのみ可能です。リスト中でクリックやダブルクリックしても (たとえ**入力可**オプションが選択されていても) 何も効果はありませんが、On ClickedやOn Double Clickedイベントは生成されます。このモードでは、開発者が (ハイライトセットを使用して) セレクションに関して、(EDIT ITEMコマンドを使用して) データ

入力に関して完全なコントロールを持ちます。On Selection ChangeとOn Before Data Entryイベントは生成されません。他方On After Editイベントはユーザーが[EDIT ITEM](コマンドを通してデータを入力した際に生成されます。
  - **単一**: 一度に一行のみ選択できます。
  - **複数**: 標準のショートカットを使用して複数行を選択できます。**Shift+クリック**で連続した行を、**Ctrl+クリック** (Windows) や **Command+クリック** (Mac OS) で非連続行を選択できます。
- **行をダブルクリック**(セレクション型のリストボックスのみ): ユーザーがリストボックスの列をダブルクリックした際に実行されるアクションを設定します。選択可能なオプションは以下の通りです:
  - **何もしない**(デフォルト): 行をダブルクリックしても自動アクションは何も発動しません。
  - **レコード編集**: 行をダブルクリックすると、リストボックスに対して定義された詳細フォーム内の対応するレコードが表示されます(以下の「詳細フォーム」セクションを参照して下さい)。レコードは読み書き可能モードで開かれるので、編集が可能です。
  - **レコード表示**: レコード編集と同様の挙動をしますが、レコードは読み出し専用モードで開かれるので編集はできません。

  **注:** 空の行へのダブルクリックは無視されます。
  選択されているアクションに関わらず、On Double clicked フォームイベントが生成されます。
  「レコード編集」・「レコード表示」のアクションに関してはOn Open Detail フォームイベントも生成されます。リストボックスに関連付けられた詳細フォームに表示されたレコードが閉じられる際にはOn Close Detail フォームイベントが生成されます(レコードが編集されたかどうかは問いません)。

- **詳細フォーム**(セレクション型のリストボックスのみ): リストボックスの個々のレコード編集・表示の際に使用するフォームを指定します。指定されたフォームは以下のタイミングで表示されます:
  - リストボックスに関連付けられている**サブレコード追加**または**サブレコード編集**の標準アクションを使用したとき([標準アクションの使用]を参照して下さい)。
  - **行をダブルクリック**のプロパティが「レコード編集」か「レコード表示」になっている行をダブルクリックしたとき(前述の「行をダブルクリック」のセクションを参照して下さい)。

#### 横スクロールしない列とドラッグしない列

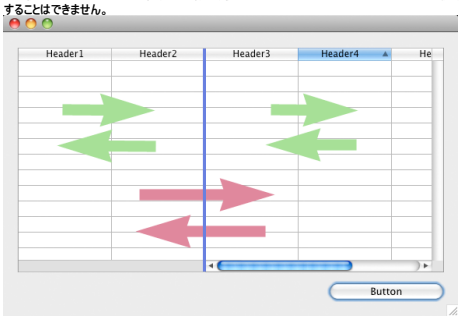
リストボックスの横スクロールしない列とドラッグしない列はそれぞれ独立して動作します:

- 横スクロールしない列は常にリストボックスの左側に表示されます。横スクロールされません。
- ドラッグしない列はリストボックス中でドラッグ＆ドロップによる列の移動ができません。

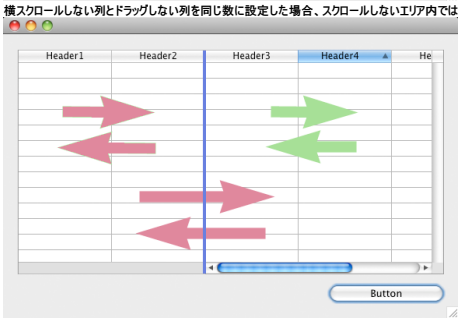
**注:** 横スクロールしない列とドラッグしない列を設定できます。詳細は4Dランゲージリファレンスの[リストボックス]を参照してください。

これらのプロパティは以下のように相互作用します:

- 列をドラッグしないのみに設定した場合、その列は移動することができません。
- 列を横スクロールしないのみに設定した場合、横スクロールしないエリア内ではドラッグで列を移動することができます。しかしエリアを越えて移動することはできません。



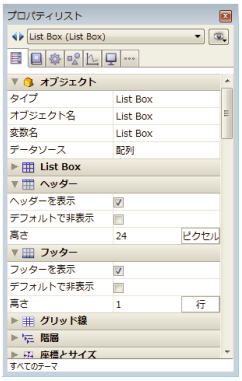

- 横スクロールしない列とドラッグしない列を同じ数に設定した場合、スクロールしないエリア内ではドラッグで移動することもできません。




- 必要に応じて横スクロールしない列数とドラッグしない列数をそれぞれ設定できます。例えば横スクロールしない列を 3、ドラッグしない列を 1に設定した場合、ユーザーは横スクロールしないエリア内で右側 2 つの列を入れ替えることができます。

#### ヘッダーとフッターテーマ

これらのテーマではリストボックス列のヘッダーとフッターの表示と高さを管理します。ヘッダーとフッターにはフォーム実行中、ユーザーが直接入力を行うことはできません。表示内容は常に計算で求められます。




- **ヘッダーを表示** / **フッターを表示**: 列ヘッダーやフッターの表示/非表示を設定します。このプロパティを選択していない場合、このエリア内のその他のプロパティを設定することはできません。
  リストボックス中でこれらのエリアが表示されたら、クリックして選択し、プロパティリストから個別のプロパティを設定できます ([リストボックスヘッダー特有のプロパティ]と[リストボックスフッター特有のプロパティ]参照)。列には1つのヘッダー1つのフッターを表示できます。それぞれは個別に設定できます。
- **デフォルトで非表示**: 他のフォームオブジェクトと同様、この設定を使用して[OBJECT SET VISIBLE]コマンドを使用したオブジェクトの表示に関する制御を容易にできます。
  [OBJECT SET VISIBLE]コマンドは対応する"ヘッダーを表示"や"フッターを表示"オプションがリストボックスのプロパティリストで選択されていない場合、効果がありません。
- **高さ**: ヘッダーやフッターの高さを設定するために使用します。まず高さの単位を設定できます (**行**または**ピクセル**)。この点に関する詳細は高さの単位 (ピクセルまたは行)を参照してください。
  デフォルトでヘッダーとフッターの高さは**1行**です。
  **警告**: ピクセル単位でのヘッダーの最小高さはシステムにより異なります。最小値よりも小さな値を指定した場合、最小値で置き換えられます。行とフッターには最小値がありません。
  **互換性に関する注意**: Windows 7とWindows Vistaではヘッダーの最小高さは24ピクセルです。変換されたデータベースで高さがこれよりも小さな値に設定されていた場合、自動でリサイズされます。この場合、それに応じてフォームを変更する必要があるかもしれません。

  ヘッダーやフッターの高さは[LISTBOX SET HEADERS HEIGHT]や[LISTBOX SET FOOTERS HEIGHT]コマンドを使用して設定することもできます。

#### 高さの単位 (ピクセルまたは行)

ヘッダー、フッター、および行の高さを**ピクセル**または**テキスト行**単位で設定できます。プロパティリストで高さフィールドのボタンを使用して単位を変更できます。

| | | |
|---|---|---|
| 高さ | 1 | 行 |

または高さを入力するエリアに直接単位を入力することもできます。行単位の場合は "L"、ピクセル単位の場合は "P" を入力します ( 例えば "17 P")。ボタンラベルは入力に基づき更新されます。

同じリストボックス内でヘッダーとフッター異なる単位を使用することもできます:

- "ピクセル" を使用した場合、指定された値は関連する行に対し直接適用されます。列に使用されるフォントサイズ等は考慮されません。フォントが行の高さに対して大きい場合、テキストは切り取られます。ピクチャーはフォーマットに基づき切り取られるかリサイズされます。
- "行" を使用した場合、高さは関連する行のフォントサイズに合わせて計算されます。複数の異なるサイズが設定されている場合、4D はもっとも大きなものを使用します。
  例えば行に "Verdana 18", "Geneva 12" そして "Arial 9" が設定されている場合、4D は行の高さの決定に "Verdana 18" を使用します。
  複数行の場合はこの高さの倍数が使用されます。
  **注：**この計算にはピクチャーのサイズやフォントに適用されるスタイルは考慮されません。

**単位の変換**: 単位を変更した場合、4D は自動で値を再計算し、結果をプロパティリストに表示します。例えば使用されるフォントが "Lucida grande 24" で高さが "1 行 " に設定されていれば "30 ピクセル " に、"2 行" なら "60 ピクセル " になります。
単位の変更を繰り返すと、4D が自動で計算を行うため、最初の値とは結果が異なってしまうこともあります。例えば以下のようになります :
(Arial 18): 52 ピクセル -> 2 行 -> 40 ピクセル
(Arial 12): 3 ピクセル -> 0.4 行が 1 行に変換される -> 19 ピクセル

#### グリッドラインテーマ

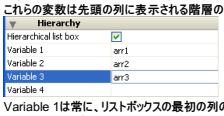
このテーマにはリストボックス中に表示されるグリッドラインに関連するプロパティが集められています。

- **水平線**: リストボックスの水平線の表示/非表示を設定します (デフォルトで表示)。
- **水平線カラー**: リストボックスの水平線のカラーを設定します (デフォルトでグレー)。
- **垂直線**: リストボックスの垂直線の表示/非表示を設定します (デフォルトで表示)。
- **垂直線カラー**: リストボックスの垂直線のカラーを設定します (デフォルトでグレー)。

#### 階層テーマ

このテーマはリストボックスのデータソースが**配列**である場合にのみ利用できます。この設定を使用してリストボックスの階層表示を設定します。リストボックスオブジェクトのポップアップメニューを使用して階層を定義すると、自動でこれらのプロパティは更新されます (階層リストボックスの作成と利用参照)。

- **階層リストボックス**: リストボックスを階層形式で表示する場合、選択します。
- **Variable 1 ... 10**: これら追加のオプションは**階層リストボックス**オプションが選択されると表示されます。入力欄に値が入力されると、新しい入力欄が追加されます。10個までの変数を指定できます。
  これらの変数は先頭の列に表示される階層のレベルを設定します。




  Variable 1は常に、リストボックスの最初の列の変数名に対応します (2つの値が自動でバインドされます)。このVariable 1欄は常に表示され、入力できます。例: country
  Variable 2も常に表示され、入力できます。これは二番目の階層レベルを指定します。例: regions
  三番目以降の欄は、その前の番号の欄が入力されると表示されます。例えば: counties、cities等。最大10レベルまで指定できます。
  ある階層レベルの値を削除すると、そのあとの階層レベルが調整されます。
  最後の変数に複数の同じ値が存在しても、この変数が階層になることはありません。例えば*arr1*にA A A B B B、*arr2*に1 1 1 2 2 2、そして*arr3*にX X Y Y Y Zが値として設定されています。この場合、A、B、1、そして2は階層で表示できますが、XとYは階層になりません:
  + A
    + 1
      X
      X
      Y
  + B
    + 2
      Y
      Y
      Z
  この原則は階層がひとつだけ設定されている場合には適用されません。この場合、同じ値はグループ化されます。
  **注：**既存のリストボックスで階層を設定した場合、(最初のものを除き) これらの列を削除または隠さなければなりません。そうしないとそれらはリストボックス中で重複して表示されます。エディターのポップアップメニューを使用して階層を設定すると (階層リストボックスの作成と利用参照)、不要な列は自動でリストボックスから取り除かれます。

#### 位置調整＆サイズ調整テーマ

このテーマにはリストボックスの座標や幅、高さに関するプロパティが集められています。

**行の高さ**プロパティは特別です。このプロパティはリストボックス行の高さを設定します。ヘッダーやフッターの高さはそれぞれのプロパティリストテーマで別に設定します。

高さは**ピクセル**または**行**単位で設定できます。

単位の選択については上記"高さの単位 (ピクセルまたは行)"を参照してください。

デフォルトで行の高さはプラットフォームとフォントサイズに基づき設定されます。

#### 背景色と境界線テーマ

このテーマには行の背景色と境界線スタイルに関連するプロパティが集められています。

- **背景色**: リストボックスの背景色を設定します。この色はヘッダー部を除くオブジェクト船体に使用されます。
- **奇数行背景色**: リストボックスの奇数行に異なるカラーを設定するために使用します。奇数行背景色を設定することで、配列を読みやすくなります。
- **境界線スタイル**: リストボックスの境界線のスタイルを設定します。

#### テキストテーマ

このテーマにはリストボックスに表示されるテキストの定義に関する標準のプロパティ(スタイルシート、フォント、スタイル)がグループ化されています。また、リストボックスのセル毎のテキスト表示の設定に使用されるいくつかのプロパティがあります:

- **行スタイル配列** (配列タイプリストボックス) / **行スタイル** (セレクションタイプリストボックス): リストボックスの各行にカスタマイズされた文字スタイルを適用するために使用します。
  - 配 列タイプのリストボックスの場 合、倍長整数配列名を入力します。この配列の各要素がリストボックスの行に対応します。この配列のサイズは列に割り当てた配 列と同じでなければなりません。(メソッドを使用して) 配列を埋めるためには、"Font Styles"テーマの定数を使用します。定数を加算することでスタイルを合成できます。リストボックスのプロパティで設定したスタイルを行に適用するた めには、配列の対応する要素に値-255を渡します。
  - セレクションタイプのリストボックスの場合、式または変数 (配列を除く) を入力します。式や変数は各行が表示される際に評価されます。式を指定する際にフォーミュラエディターを使用することができます。これを行うには、エリア を選択したときに表示される[...]ボタンをクリックします。"Font Styles"テーマの定数を使用することができます。
    以下の例は変数名を使用しています。**行スタイル**エリアに*CompanyStyle*と入力され、フォームメソッドにOn Display Detailで実行される以下のコードが書かれています:
    ```
    CompanyStyle:=Choose([Companies]ID;Bold;Plain;Italic;Underline)
    ```
- **行フォントカラー配列** (配列タイプリストボックス) / **行フォントカラー** (セレクションタイプリストボックス): リストボックスの各行にカスタマイズしたフォントカラーを適用するために使用します。RGBカラーを使用しなければなりません。この点に関する詳細は4D ランゲージリファレンスマニュアルのOBJECT SET RGB COLORSコマンドの説明を参照してください。
  - 配列タイプのリスト ボックスの場 合、倍長整数配列名を入力します。この配列の各要素がリストボックスの行に対応します。この配列のサイズは列に割り当てた配列と同じでなけれ ばなりません。"SET RGB COLORS"テーマの定数を使用できます。リストボックスプロパティで設定したフォントカラーを行に適用するには、配列の対応する要素に値-255を渡 します。
  - セ レクションタイプのリストボックスの場合、式または変数 (配列を除く) を入力します。式や変数は各行が表示される際に評価されます。式を指定する際にフォーミュラエディターを使用することができます。これを行うには、エリア を選択したときに表示される[...]ボタンをクリックします。"SET RGB COLORS"テーマの定数を使用することができます。
    以下の例は変数名を使用しています。**行フォントカラー**エリアに*CompanyColor*と入力され、フォームメソッドにOn Display Detailで実行される以下のコードが書かれています:
    ```
    CompanyColor:=Choose([Companies]ID;Default background color;Default light shadow color;Default
    foreground color;Default dark shadow color)
    ```
- **行背景色配列** (配列タイプリストボックス) / **行背景色** (セレクションタイプリストボックス): リストボックスの各行にカスタマイズされた背景色を適用するために使用します。RGBカラー値を使用しなければなりません。この点に関する詳細は4Dラン ゲージリファレンスマニュアルのOBJECT SET RGB COLORSコマンドの説明を参照してください。
  - 配列タイプのリストボックスの場 合、倍長整数配列名を入力します。この配列の各要素がリストボックスの行に対応します。この配列のサイズは列に割り当てた配列と同じでなければなりまし ん。"SET RGB COLORS"テーマの定数を使用できます。リストボックスプロパティで設定した背景色を行に適用するには、配列の対応する要素に値-255を渡します。
  - セレクションタイプのリストボックスの場合、式または変数 (配列を除く) を入力します。式や変数は各行が表示される際に評価されます。式を指定する際にフォーミュラエディターを使用することができます。これを行うには、エリア を選択したときに表示される[...]ボタンをクリックします。"SET RGB COLORS"テーマの定数を使用することができます。

- **非表示行配列** (配列タイプリストボックスのみ): リストボックスの任意の行を非表示にするために使用します。ブール配列名を入力しなければなりません。この配列サイズはリストボックスの行数と同じでなけ ればなりません。配列の各要素はリストボックス行の表示/非表示ステータスを表します:
  - 行を隠すにはTrueを渡します。
  - 行を表示するにはFalseを渡します。

  配列が指定されていないか、配列が空の場合、リストボックスのすべての行が表示されます。デフォルトでつきあされた行は表示されます。
  行 の非表示はリストボックスの表示にのみ影響します。非表示行は依然として配列に存在し、プログラムで管理可能です。ランゲージコマンド、特にLISTBOX Get number of rowsやLISTBOX GET CELL POSITIONはリストボックス行の表示/非表示の影響を受けません。例えば10行あるリストボックスで先頭の9行が隠されている場合、LISTBOX Get number of rowsは10を返します。

ユーザーの視点から見た場合、リストボックス中に非表示行があることは視認できません。表示されている行だけが選択できます (例えばすべてを選択コマンドを使用して)。

- **選択モード**: リストボックス行の選択モードを指定するために使用します。3つのモードから選択できます:
  - **なし**: 行を選択することはできず、データを入力することもできません。選択とデータ入力はプログラムからのみ可能です。リスト中でクリックやダブルクリックしても (たとえ**入力可**オプションが選択されていても) 何も効果はありませんが、On ClickedやOn Double Clickedイベントは生成されます。このモードでは、開発者が (ハイライトセットを使用して) セレクションに関して、(EDIT ITEMコマンドを使用して) データ入力に関して完全なコントロールを持ちます。On Selection ChangeとOn Before Data Entryイベントは生成されません。他方On After Editイベントはユーザーが EDIT ITEMコマンドを通してデータを入力した際に生成されます。
  - **単一**: 一度に一行のみ選択できます。
  - **複数**: 標準のショートカットを使用して複数行を選択できます。**Shift+クリック**で連続した行を、**Ctrl+クリック** (Windows) や **Command+クリック** (Mac OS) で非連続行を選択できます。
- **横揃え**: セルの内容を**左**、**中央**、または**右**に揃えるために使用されます。
  **デフォルト**オプションを選択した場合、各列のデータ型に基づき整列方向が決定されます。テキストやピクチャーは左、数値は右です。
- **縦揃え**: セルの内容を**上**、**中央**、または**下**に揃えるために使用されます。
  **デフォルト**オプションを選択した場合、各列のデータ型に基づき整列方向が決定されます。ピクチャーを除き、すべて下です。ピクチャーは上です。

整列プロパティはリストボックスレベルで全体に、またはそれぞれの列ごとに個々に適用できます(スタイルとカラーの管理を参照して下さい)。さらに、横揃え/縦揃えのプロパティはヘッダー/フッターエリアにも設定できます。

#### アクションテーマ

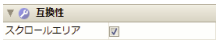
このテーマではリストボックスの動作に関連するプロパティを設定します。

- **メソッド** (編集...): このボタンをクリックすると、リストボックスオブジェクトのメソッドがメソッドエディターで開かれます (各列にもオブジェクトメソッドを記述できる点に留意してください)。
- **ドラッグ可**と**ドロップ可**: リストボックスのドラッグ＆ドロップ機能を有効にします。これによりリストボックス行と他のリストボックスあるいは他の4Dオブジェクト間のドラッグ＆ドロップが可能になります。リストボックスの行だけが関連します。列をドラッグ＆ドロップすることはできません (リストボックス内で列を移動することはできます)。
  リストボックス内でのドラッグ＆ドロップは4D標準のドラッグ＆ドロップメカニズムを使用して管理されます (On DropとOn Drag Overフォームイベント、DRAG AND DROP PROPERTIESとDrop positionコマンド)。
- **行の移動可** (配列タイプのリストボックスのみ): 実行時に行の移動を許可します。このオプションはデフォルトでチェックされています。セレクションタイプや (**階層リストボックス**オプションがチェックされている) 階層リストボックスではこのオプションを使用できません。
- **並び替え可**: ヘッダーのクリックによる列データの並び替えを有効にします。このオプションはデフォルトで選択されています。
  ピクチャー型配列 (列) はこのメカニズムではソートできません。
  レコードのセレクションに基づくリストボックスの場合、標準の並び替え機能は以下の場合のみ有効です:
  - データソースが**カレントセレクションであり、**
  - その列にフィールドが割り当てられていること (文字、数値、日付、時間、およびブール型)。

  他の場合 (命名セレクションに基づくリストボックスや式が割り当てられた列)、標準の並び替え機能は動作しません。
  標準のリストボックス並び替えは、データベースのカレントセレクションの順番を変更します。しかしハイライトされたレコードと、カレントレコードは変更されません。
  標準の並び替えは、式が割り当てられた列も含め、リストボックスのすべての列を同期します。

#### 互換性テーマ

このテーマにある**スクロールエリア**オプションを使用して、以前のスクロールエリアの動作をエミュレートできます。以前のスクロールエリアはv13以降に変換すると自動でリストボックスに変換されます。なおこの互換性チェックボックスはすべてのリストボックスで利用できます:




リストボックスでこのオプションが選択されていると以下の特別な動作を行うようになります:

- リストボックス列配列に非表示プロパティが設定されていると、リストボックス全体が非表示になります。
- リストボックス列の配列変数に数値を代入すると、当該行が選択されます ( 例 : MyArray:=5 とするとリストボックス中 5 番目の要素が選択されます )。
- 逆に行をクリックすると列配列の現在値が更新されます。
- リストボックス行から他のオブジェクトにドロップが行われたとき、このオブジェクト内で実行されるDRAG AND DROP PROPERTIESコマンドは (リストボックス自身ではなく) リストボックス列への配列を返します。

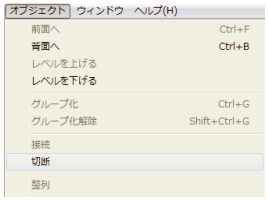
##### 接続されたリストボックス

グループ化されたスクロールエリアがリストボックスに変換されると、それらは接続されます。接続されたリストボックスは連携して動作します:

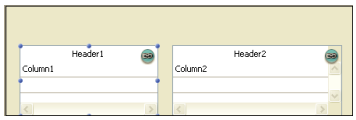
- ひとつのリストボックス上で行を選択すると、接続されたグループに属するすべてのリストボックスの同じ行が選択されます。
- リストボックスをスクロールすると、接続されたグループに属するすべてのリストボックスがスクロールされます。

**注:** 変換されたリストボックスはフォーム上で**グループ化**されています。

フォームエディターのオブジェクトメニューから、新しい接続および切断メニューを使用して、これらのリストボックスの接続 / 切断を切り替えることができます:



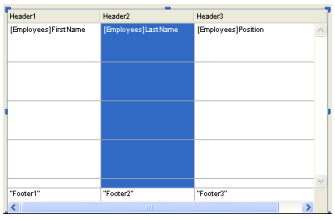

これらのコマンドはフォームエディター上で適切なリストボックスが選択されている場合に有効となります。接続されたリストボックスが選択されている場合、そのリストボックスに接続されたすべてのリストボックス上に特別なバッジが表示されます :




**互換性に関する注意**: この動作原理を用いて、グループ化されたスクロールエリアの動作を再現します。しかしこの動作を標準のリストボックスに置き換えることを推奨します。

## リストボックス列特有のプロパティ

リストボックスオブジェクトが選択されているときにリストボックスレッツをクリックすると、その列を選択することができます:

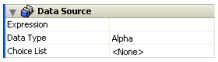



リストボックス列ごとに標準のプロパティ (テキスト、背景色等) を設定できます。ここで設定されたプロパティは、リストボックスオブジェクトで設定されたものよりも優先されます。

この節ではその他のリストボックス列に特有のプロパティについて説明します。

### データソーステーマ

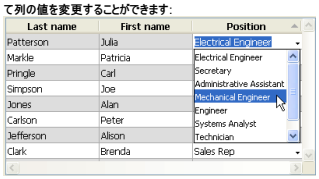
- **式** (セレクションタイプリストボックス): 列に割り当てる4D式を、式欄で指定します:




- 以下を指定できます:
  - *4D式* (単純な式、フォーミュラ、または4Dメソッド)。式の結果はアプリケーションモードでフォームを実行すると自動で表示されます。式はマスターテーブルの (カレントまたは命名) セレクションの各レコード毎に評価されます。セレクションが空の場合、列には何も表示されません。
  - *変数* (この場合、コンパイルするためには、明示的に宣言しなければなりません)。Blobや配列以外、どの型の変数でも使用できます。変数の値は通常On Display Detailイベントで計算されます。
  - 標準の[Table]Fieldシンタックスを使用したフィールド (例: [Employees]LastName)。以下の型のフィールドを使用できます:
    - 文字
    - テキスト
    - 数値
    - 日付
    - 時間
    - ピクチャー
    - ブール

  マスターテーブルおよび他のテーブルのフィールドを指定できます。
  いずれの場合もプロパティリストの[...]ボタンをクリックして4Dフォーミュラエディターを開き、式を指定することができます。式を使用する場合、**入力可**オプションが選択されていても、その列に値を入力することはできません。
  フィールドや変数を使用する場合、**入力可** (とストラクチャーエディターの"更新不可"属性) の設定に基づき列への入力可/不可が決定されます。
  デザインモードでは、列の最初の行にデータソースのタイプが表示されます。例えば*Field=[Table1]MyFld*のように表示されます。
  指定された式が正しくない場合、アプリケーションモードでその列にはエラーメッセージが表示されます。

- **デフォルト値** (配列タイプリストボックス): 列のデフォルト値のリストを入力するためのダイアログボックスを表示すすには**編集...**ボタンをクリックします。入力した値は自動で、フォームを実行したときにその列に割り当てられた配列変数に代入されます。
  各要素は改行で区切ります。このダイアログボックスに関する詳細は[データ入力の制御と補助](#)のデフォルト値を参照してください。
- **データタイプ** (セレクションタイプリストボックス): このメニューを使用して列に割り当てられた式や変数の型を指定します。これは適用する表示フォーマットを示し、"表示"テーマの**表示タイプ**メニューを更新するために使用します。
  式エリアにフィールドが入力されると、**データタイプ**プロパティは表示されません。フィールド型に対応した表示フォーマットが使用されます。
- **選択リスト**: このプロパティを使用してリストボックス列に選択リストを割り当てることができます。リストを使用すると、ポップアップメニューを使用して列の値を変更することができます:




  依然キーボードからの入力も可能である点に留意してください。選択リストを使用してのみ、値の変更を許可したい場合は、指定リストも選択します (後述)。
  **注:** 階層リストを指定した場合、第一レベルだけが利用できます。
- **関連付け(リスト項目の値/参照番号)**: 列が選択リストと関連付けられているとき、このプロパティによって列のフィールドまたは変数の中に保存されるコンテンツの型を指定することが出来ます。このオプションについての詳細は、[関連付(値または参照番号)](#)を参照して下さい。

### 位置調整＆サイズ調整

このテーマにはリストボックス列の幅に関するプロパティが集められています。

- **幅**: デフォルトの列幅 (ピクセル単位)。この値はフォームエディターでマウスを使用して列をリサイズすると更新されます。
  **サイズ変更可**プロパティが選択されていると、ユーザーがマウスで列幅を変更することもできます。
- **最小幅**: 列の最小幅 (ピクセル単位)。列やフォームをサイズ変更する際、列幅はこのサイズよりも小さくすることができません。
- **最大幅**: 列の最大幅 (ピクセル単位)。列やフォームをサイズ変更する際、列幅はこのサイズよりも大きくすることができません。

**注:** フォームをリサイズする際、リストボックスの水平サイジングに**拡大**が設定されていると、一番右の列は必要に応じて最大サイズを超えて拡大します。

### リサイズオプションテーマ

このテーマには (デフォルトで選択されている)**リサイズ可**オプションのみがあります。このオプションが選択されていると、ユーザーはヘッダーエリアの右側をドラッグすることで列のサイズを変更できます。

### 入力テーマ

このテーマにはリストボックス列へのデータ入力に関するプロパティがまとめられています。

- **入力可**: 列への入力を許可します (デフォルトで選択)。入力モードに入るには、ユーザーはセル上で2回クリックします。
  このプロパティを選択しない場合、"データソース"や"値の範囲"テーマにある、値を選択するためのリストは選択できなくなります。
  **注:** リストボックスへのデータ入力のために実装されたメカニズムについては4Dランゲージリファレンスマニュアルの入力の管理を参照してください。
- **入力フィルター**: 列のセルに入力フィルターを割り当てます。**入力可**プロパティが選択されていない場合、このプロパティは表示されません。
- **ワードラップ**(テキスト型の列のみ): このオプションは列の中身がカラムの幅を超えた時の表示の仕方を管理します。
  このオプションがチェックされていると、テキストがカラムの幅を越えたときに、カラムの高さが許容する範囲内で自動的に次の行へと改行します。
  このオプションがチェックされていない場合、長すぎるテキストは全て切り落とされ、省略記号(...)が表示されます。
  以下の例では、左の列ではワードラップのオプションがチェックされていて、右の列ではされていません:

| Header1 | Header2 |
|---|---|
| The vertical alignment will be applyed as long as the full text fits in the cell. Otherwise, the text will be aligned to the top so as the beginning of the text will visible. | The vertical alignment will be ap... |
| You can make a field invisible in the Application environment and for the plug-ins by selecting the Invisible property for this field. | You can make a field invisible in ... |

  **ワードラップ**のオプションの値に関わらず、行の高さは変化しないことに注意して下さい。改行を含むテキストがカラムの中に全て表示しきれないとき、表示しきれない部分は切り落とされ、省略記号も表示されません。単一の行を表示するリストボックスの場合、テキストの最初の行のみ表示されます:

| Header1 | Header2 |
|---|---|
| The vertical alignment will be | The vertical alignment will be ap... |
| You can make a field invisible in | You can make a field invisible in ... |

### 値の範囲テーマ

このテーマではリストボックス列への入力を管理するためのリストを設定します。

- **指定リスト**: 列に入力することができる値のリストを指定するために使用します。各セルに割り当てられたポップアップメニューを使用してリストの値にアクセスできます (**データソーステーマ**の"選択リスト"プロパティ参照)。
  **選択リスト**プロパティと異なり、指定リストが指定されると、キーボードからの入力はできなくなります。ポップアップメニューを使用した値の選択のみが可能となります。**選択リストと指定リスト**プロパティで異なるリストが選択されると、**指定リスト**プロパティが優先されます。
- **除外リスト**: 列に入力することができない値のリストを指定するために使用します。除外された値が入力されると受け入れられず、エラーメッセー

ジが表示されます。

**注:** 指定されたリストが階層の場合、第一レベルのみが使用されます。

**表示テーマ**

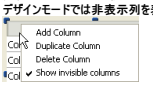
このテーマでは、列値の表示フォーマットや**非表示**プロパティの状態を設定します。このテーマの内容はオブジェクトテーマで設定された変数の型によって変化します。

**注:** リストボックスが階層モードで表示される際、第一列目において、このテーマのすべてのプロパティはすべて設定不可となります。

- **タイプフォーマット**: 列のデータに表示フォーマットを割り当てるために使用します。提供されるフォーマットは変数型 (配列タイプのリストボックス) またはデータ/フィールド型 (セレクションタイプのリストボックス) により異なります。使用できる標準の4Dフォーマットは文字、数値、日付、時間、ピクチャー、そしてブールです。テキスト型に特別な表示フォーマットはありません。既存のカスタムフォーマットを使用することもできます。
- ブール配列をチェックボックスやポップアップメニューとして表示することもできます。**チェックボックス**オプションを選択すると**タイトル**プロパティが表示され、チェックボックスのタイトルを設定できます。**ポップアップ**オプションを選択すると、**テキスト (True時)** と**テキスト (False時)** プロパティが表示され、ポップアップメニューの対応するタイトルを設定できます。
- 数値を表示する列は3ステートチェックボックスとして表示できます。このオプションは**表示タイプ**ドロップダウンリストにあります。**3ステートチェックボックス**タイプを選択すると、以下の値が表示されます:
  - 0 = チェックされていない
  - 1 = チェックされている
  - 2 (または2以上の任意の数値) セミチェックボックス (三番目の状態)、データ入力時、この状態は2を返します。
  - -1 = 非表示チェックボックス
  - -2 = チェックされていない、入力不可
  - -3 = チェックされている、入力不可
  - -4 = セミチェックボックス、入力不可

  この場合も**タイトル**プロパティが表示され、チェックボックスのタイトルを指定できます。

- **非表示**プロパティが選択されていると、アプリケーションモードで列が隠されます。
  デザインモードでは非表示列を表示するか隠すかを、リストボックスのコンテキストメニューで選択できます (列または列ヘッダー上で右クリック):




**背景色と境界線テーマ**

プロパティリストのこのテーマでは、列のセルの背景色を設定します。

- **背景色**: 列の背景色を設定することが出来ます。デフォルトでは、**自動**が選択されています。列は、この背景色をリストボックスレベルで使用します。
- **交互に使用する背景色**: 列の中の奇数番の行に使用するための異なる背景色を設定することが出来ます。デフォルトでは、**自動**が選択されています。列は、この交互に使用する背景色をリストボックスレベルで使用します。
- **行背景色配列**(配列型リストボックス)/**背景色**(セレクション型リストボックス): 列のそれぞれのセルにカスタムの背景色を指定することが出来ます。指定するにはRGBカラー値を使用します。
  - 配列型のリストボックスの場合には、倍長整数型配列の名前を入力しなければなりません。この配列のそれぞれの要素は列のそれぞれのセルと対応するので、二つの配列のサイズは同じである必要があります。この背景色を指定するときに"SET RGB COLORS"テーマ内の定数を使用することができます。もし上のレベルで定義されている背景色をそのまま継承したい場合 (継承を参照して下さい)には、対応する配列の要素に -255 を渡して下さい。
  - セレクション型リストボックスの場合には、4D式か変数(ただし配列型を除く)を入力しなければなりません。4D式または変数は、それぞれの表示されているセルごとに評価されます。フォーミューラーエディターを使用して4D式を編集することが可能です。そのためにはエリアを選択したときに表示される [...] ボタンをクリックして下さい。"SET RGB COLORS"テーマ内の定数を使用することができます。

**テキストのテーマ**

- **行スタイル配列**(配列型リストボックス)/**スタイル**(セレクション型リストボックス):列のそれぞれのセルにカスタムのスタイルを指定することが出来ます。
  - 配列型のリストボックスには、倍長整数型配列の名前を入力しなければなりません。この配列のそれぞれの要素は、列のそれぞれのセルと対応するので、二つの配列のサイズは同じである必要があります。配列へは、"Font Styles"テーマ内の定数を使用することができます(メソッドを使用しての入力も可能)。定数同士を足し合わせてスタイルを組み合わせることもできます。もし上のレベルで定義されているスタイルをそのまま継承したい場合(継承を参照して下さい)には、対応する配列の要素に -255 を渡して下さい。
  - セレクション型リストボックスの場合には、4D 式か変数(ただし配列型を除く)を入力しなければなりません。4D 式または変数は、それぞれの表示されているセルごとに評価されます。フォーミューラーエディターを使用して 4D 式を編集することが可能です。そのためにはエリアを選択したときに表示される **[...]** ボタンをクリックして下さい。"Font Styles" テーマ内の定数を使用することができます。
- **行フォントカラー配列**(配列型リストボックス)/**フォントカラー**(セレクション型リストボックス): 列のそれぞれのセルにカスタムのフォントカラーを指定することが出来ます。指定するには RGB カラー値を使用しなければなりません。
  - 配列型のリストボックスの場合には、倍長整数型配列の名前を入力しなければなりません。この配列のそれぞれの要素は列のそれぞれのセルと対応するので、二つの配列のサイズは同じである必要があります。このフォントカラーを指定するときに"SET RGB COLORS"テーマ内の定数を使用することができます。もし上のレベルで定義されているフォントカラーをそのまま継承したい場合(継承を参照して下さい)には、対応する配列の要素に -255 を渡して下さい。
  - セレクション型リストボックスの場合には、4D式か変数(ただし配列型を除く)を入力しなければなりません。4D 式または変数は、それぞれの表示されているセルごとに評価されます。フォーミューラーエディターを使用して 4D式を編集することが可能です。そのためにはエリアを選択したときに表示される **[...]** ボタンをクリックして下さい。"SET RGB COLORS" テーマ内の定数を使用することができます。

## リストボックスヘッダー特有のプロパティ

**注**: リストボックスのヘッダープロパティにアクセスするためにはリストボックスのプロパティリストで"**ヘッダーを表示**"オプションが選択されていなければなりません(リストボックス特有のプロパティ参照)。

フォームエディターでリストボックスオブジェクトが選択されているときに、リストボックスヘッダーをクリックするとヘッダーを選択できます:

| 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} |

リストボックスの各列ヘッダー毎に標準のテキストプロパティを設定できます。この場合、これらのプロパティはリストボックスや列に設定されたプロパティよりも優先されます。

さらにこの節で説明する特定のプロパティを設定することができます。

**オブジェクトテーマ**

このテーマにはヘッダーを定義するために使用されるプロパティが含まれています。

- **オブジェクト名**: ヘッダーオブジェクトの名前。
- **変数名**: ヘッダーオブジェクトに割り当てる変数の名前。この変数は数値型であり、列の並び替えや並び替えを示す矢印の表示をプログラムで管理するために使用されます (4Dランゲージリファレンスマニュアルのリストボックスオブジェクトの管理参照)。
- **タイトル**: ヘッダーに表示されるラベル。

**ピクチャーテーマ**

このテーマには、列ヘッダーにピクチャーを表示するためのプロパティが含まれています (設定は任意)。アイコンは特にカスタマイズされた並び替えを行う際などに、列タイトルの隣、あるいはタイトルの場所に表示できます。

| Last name | Position |
|---|---|
| James | Engineer |

- **アイコン**: ヘッダーに挿入するピクチャーのソースを指定します。4Dピクチャーボタンのように変数、ピクチャーライブラリー、リソースファイル、またはファイルのピクチャーを使用できます。
- **名前/ID**: 指定したソース内で使用するピクチャーを指定します。ここに入力する値は選択したソースにより異なります。ピクチャーが変数の場合変数名、ピクチャーライブラリーやリソースファイルの場合ID番号、ピクチャーファイルの場合パス名を入力します。パス名はデータベースのResourcesフォルダーからの相対パスで記述します (ピクチャーファイルの自動参照参照)。
- **アイコンの場所**: ヘッダー中でのアイコンの場所。ヘッダーの**左**または**右**を選択できます。
- **ミラー効果 (Windows)**: アイコンを右から左モードでして表示するために使用します (ミラー効果 (Windows)参照)。

**ヘルプテーマ**

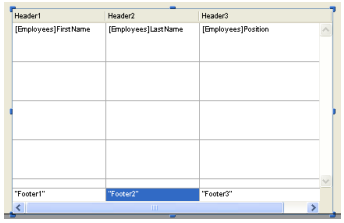
各ヘッダーは個別にヘルプメッセージを持つことができます。

## リストボックスフッター特有のプロパティ

リストボックスフッターはユーザーによる入力ができず、追加の情報を表示するために使用されます。表形式で表示されるデータに対し、通常フッターは合計や平均などの計算値を表示するために使用されます。

**注:** リストボックスのフッタープロパティにアクセスするためには、リストボックスのプロパティリストでフッターを表示が選択されていなければなりません (リストボックス特有のプロパティ参照)。

リストボックスへのフッター表示設定を行ったら、フッターセルをクリックして選択し、プロパティリストからさらに詳細な設定を行うことができます:




各リストボックス列フッターに標準のテキストプロパティを設定できます。設定するとリストボックスや列に対する設定よりもこちらが優先されます。

そのほかにこのページで説明するプロパティにアクセスできます。

**オブジェクトテーマ**

このテーマにはフッター定義のプロパティが含まれます。

- **オブジェクト名と変数名**: フッターエリアはフォーム内でユニークなオブジェクト名、および変数を割り当てることができます。デフォルトで変数名は空であり、4D はダイナミック変数を使用します。
- **変数タイプ**: このメニューで変数の型を設定し、またプロパティリストの他のオプション表示を更新します。ダイナミック変数を使用しない場合、変数の型定義はランゲージを使用して行わなければなりません。
- **変数の計算**: このオプションを使用してフッターエリアに適用される計算タイプを設定します。いくつかの計算オプションの他、カスタムも選択できます:
  - **最小値** , **最大値** , **合計値** , **カウント** , **平均値**, **標準偏差**(*) , **分散値**(*)および**平方和**(*)。これらの計算は " 自動計算 " で説明されています。計算が選択されると、リストボックス列中のすべての値が自動で計算の対象となります。リストボックス行の表示 / 非表示状態は考慮されません。表示行だけを計算対象にしたい場合、カスタムを選択してプログラムコードで計算します。
    フッターエリアに自動計算が設定されるとフォームエディター上では"標準アクション" バッジが表示されます:

    
  - **カスタム**: このオプションを選択した場合、4D は自動計算を行いません。プログラムを使用して表示する値をエリアに代入しなければなりません。

**自動計算**

フッターエリアには様々な自動計算を割り当てることができます。以下の表は列のタイプに基づき使用することのできる計算と、(コードで明示的に宣言されていないとき) 4Dがフッター変数に自動で割り当てる型を示しています:

| | 数値 | テキスト | 日付 | 時間 | ブール | ピクチャー | *フッターの変数型* |
|---|---|---|---|---|---|---|---|
| **最小値** | ○ | | ○ | ○ | ○ | | 列の型と同じ |
| **最大値** | ○ | | ○ | ○ | ○ | | 列の型と同じ |
| **合計値** | ○ | | | ○ | ○ | | 列の型と同じ |
| **カウント** | ○ | ○ | ○ | ○ | ○ | ○ | 倍長整数 |
| **平均値** | ○ | | | ○ | | | 実数 |
| **標準偏差(*)** | ○ | | | ○ | | | 実数 |
| **分散値(*)** | ○ | | | ○ | | | 実数 |
| **平方和(*)** | ○ | | | ○ | | | 実数 |

(*) 配列タイプのリストボックスのみ

**注:** フォーミュラを割り当てた列には自動計算を関連付けることができません。この場合、カスタム計算を行わなければなりません。

**ヘルプテーマ**

各フッターエリアはヘルプメッセージを持つことができます。

## 階層リストボックスの作成と利用

リストボックスを階層表示にするよう指定することができます。階層リストボックスは左の列が階層状に表示されます。このタイプの表示方法は、繰り返される値や、階層に依存するデータの表示などに適用できます (国/地域/都市など)。

**配列タイプのリストボックス**のみを階層にできます。

階層リストボックスはデータを表示する特別な方法ですが、データの構造 (配列) は変更しません。階層リストボックスは通常のリストボックスと全く同じ方法で管理されます。

階層リストボックスと指定するには、3つの方法があります:

- フォームエディターのプロパティリストを使用して階層要素を手作業で設定する。
- フォームエディターのリストボックス管理メニューを使用して階層を生成する。
- LISTBOX SET HIERARCHYやLISTBOX GET HIERARCHYコマンドを使用する。

この節では4Dのフォームエディターで階層リストボックスを作成する方法と、実行時にそれがどのように動作するかについてその基礎を説明します。階層リストボックスのプログラムによる管理 (選択やブレーク、On ExpandやOn Collapseフォームイベントの利用方法等) については、4Dランゲージリファレンスマニュアルの階層リストボックスの管理を参照してください。

### プロパティリストを使用して階層を管理する

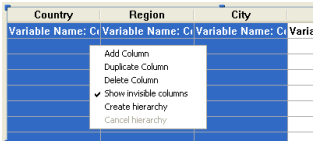
プロパティリストの"階層"テーマで階層モードを設定できます。詳細はリストボックス特有のプロパティを参照してください。

### コンテキストメニューを使用して階層を管理する

フォームエディターのリストボックスの列エリアを右クリックして表示されるコンテキストメニューから、**階層を作成**や**階層をキャンセル**コマンドにアクセスできます。

#### 階層を作成

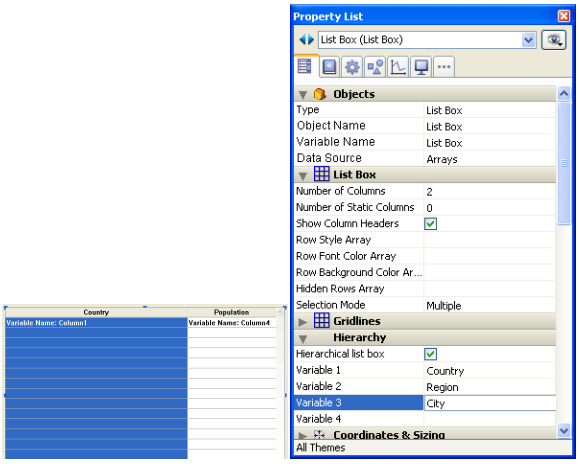
フォームエディター中で配列タイプのリストボックスオブジェクトの一番目から任意の数の列を選択すると、**階層を作成**コマンドがコンテキストメニューから利用できるようになります:




このコマンドを選択すると、以下のアクションが実行されます:

- そのオブジェクトのプロパティリストで"階層リストボックス"オプションが選択されます。
- その列の変数が階層を指定するために使用されます。既に設定されていた内容は新しい内容で置き換えられます。
- (最初の列を除き) 選択された列はリストボックス内に表示されなくなります。

例: 左から国、地域、都市、人口列が設定されたリストボックスがあります。国、地域、都市が (下図の通り) 選択され、コンテキストメニューから**階層を作成**を選択すると、最初の列に3レベルの階層が作成され、二番目と三番目の列は取り除かれます。人口列が二番目になります:




#### 階層をキャンセル

階層リストボックスとして定義されたリストボックスで最初の列を選択すると、**階層をキャンセル**コマンドを使用できます。このコマンドを選択すると以下のアクションが実行されます:

- そのオブジェクトの"階層リストボックス"オプションの選択が解除されます。
- 2番目以降の階層レベルが削除され、通常の列としてリストボックスに追加されます。

### 動作

階層リストボックスを含むフォームが最初に開かれる際、デフォルトですべての行が展開されています。

配列中で値が繰り返されていると、ブレーク行と階層"ノード"が自動でリストボックスに追加されます。例えばリストボックスに都市に関する4つの配列が含まれていて、それぞれ国、地域、都市名、人口データが含まれているとします:

| Country | Region | City | Population |
|---|---|---|---|
| France | Brittany | Rennes | 200000 |
| France | Brittany | Quimper | 80000 |
| France | Brittany | Brest | 120000 |
| France | Normandy | Caen | 75000 |
| France | Normandy | Deauville | 35000 |

リストボックスが階層形式で表示されると (先頭3つの配列が階層化されている場合)、以下のように表示されます:

| City | Population |
|---|---|
| France | |
| Brittany | |
| Rennes | 200000 |
| Quimper | 80000 |
| Brest | 120000 |
| Normandy | |
| Caen | 75000 |
| Deauville | 35000 |

階層を正しく構築するためには、事前に配列をソートしなければなりません。例えば配列中にデータがAAABBAACCの順で含まれていると、階層は以下のようになります:

> A
> B
> A
> C

階層"ノード"を展開したり折りたたんだりするには、ノード上をクリックします。ノード上を**Alt+クリック** (Windows) または **Option+クリック** (Mac OS) すると、すべてのサブ要素が同時に展開されたり折りたたまれたりします。

**ソートの管理**

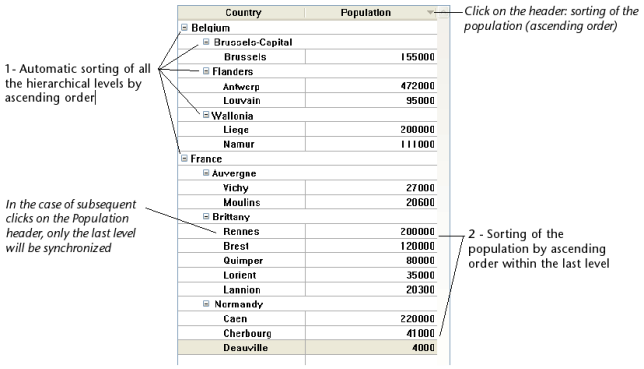
階層モードのリストボックスにおいて、(リストボックス列のヘッダーをクリックして実行される) 標準の並び替えは常に以下のように行われます:

- まず階層列 (一番目の列) のすべてのレベルが自動で昇順にソートされます。
- 次にクリックされた列の値を使用して昇順または降順にソートが実行されます。
- すべての列が同期されます。
- その後の非階層列のソート時には、一番目の最後のレベルのみがソートされます。この列のソートはそのヘッダーをクリックすることで行えます。

まだソートされていない以下のリストボックスがあります:

| Country | Population |
|---|---|
| France | |
| Brittany | |
| Rennes | 200000 |
| Quimper | 80000 |
| Brest | 120000 |
| Lannion | 20300 |
| Lorient | 35000 |
| Normandy | |
| Caen | 220000 |
| Deauville | 4000 |
| Cherbourg | 41000 |
| Auvergne | |
| Vichy | 27000 |
| Moulins | 20600 |
| Belgium | |
| Wallonia | |
| Namur | 111000 |
| Liege | 200000 |
| Flanders | |
| Antwerp | 472000 |
| Louvain | 95000 |
| Brussels-Capital | |
| Brussels | 155000 |

"Population"ヘッダーをクリックして人口に基づき昇順あるいは降順でソートを行うと、データは以下のように表示されます:




通常のリストボックスと同様、リストボックスの"ソート可"オプションの選択を解除することで標準のソートメカニズムを無効にし、プログラムでソートを管理できます。

**日付と時間の表示**

階層リストボックスに日付や時間型の値を表示する際、それらは標準フォーマットで表示されます:

- 日付はShort system formatで表示されます (例えば2009年5月30日の場合、日本語システム上では09/05/30と表示されます)。
- 時間もShort system formatで表示されます (システムパラメターに基づき"12:15:30"または"12:15"で表示されます)。

## 標準アクションの使用

ボタンまたはメニューに関連付けられた 標準アクション によって"セレクション"型のリストボックスを管理することができます。例えば、リストボックスの**サブレコード追加**の標準アクションを使ってテーブルに新しいレコードを追加することができます。この新機能によってリストボックスに基づいたモダンなリストボックスを簡単に作成できるようになります。

リストボックスでは3つの標準アクションが使用できます: **サブレコード追加**、**サブレコード編集**、そして**サブレコード削除**の3つです。

**サブレコード追加**

フォーム内に"セレクション"型のリストボックスが少なくとも一つあるとき、**サブレコード追加**の標準アクションが関連付けられているボタンまたはメニュー項目が自動的に有効になります。フォームにリストボックスが複数含まれる場合、アクションはフォーカスがあるリスト(またはデフォルトで最前面にあるリスト)に対して適用されます。

ユーザーがボタンまたはメニュー項目をクリックしたとき、リストボックスに対して定義された詳細フォームの中に新しい空のレコードが表示されます(リストボックステーマを参照して下さい)。ユーザーはここに値を入力し、確定するとまた新しい空のレコードが表示されます。これはユーザーがキャンセルボタンを押すまで繰り返されます。リストボックスのデータソースがカレントセレクションの場合、作成されたレコードは全てリストに表示されます。

**サブレコード編集**

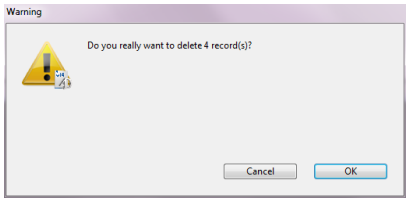
フォーム内に"セレクション"型のリストボックス内に選択されている行が少なくとも一つあるとき、**サブレコード編集**の標準アクションが関連付けられているボタンまたはメニュー項目が自動的に有効になります。複数の行が選択されたとき、このアクションは最後にセレクションに追加された行に対してのみ適用されます。

ユーザーがボタンまたはメニュー項目をクリックしたとき、リストボックスに対して定義された詳細フォームの中にクリックした行に対応するレコードが表示されます (リストボックステーマを参照して下さい)。ユーザーは値を編集でき、確定またはキャンセルするとリストボックスへと戻ります。

**サブレコード削除**

フォーム内に"セレクション"型のリストボックス内に選択されている行が少なくとも一つあるとき、**サブレコード削除**の標準アクションが関連付けられているボタンまたはメニュー項目が自動的に有効になります。複数の行が選択されたとき、このアクションは全てのレコードに対して適用されます。

ユーザーがボタンまたはメニュー項目をクリックすると、削除をするかキャンセルするかを確認するダイアログボックスが表示されます:



## リストボックスにフィールドを表示する

リストボックス列にはマスターテーブルのフィールドあるいは他のテーブルのフィールドを関連付けることができます。マスターテーブルについてはリストボックス特有のプロパティを参照してください。

すべてのケースでリストボックスの内容は、リストボックスのマスターテーブルのカレント/命名セレクションに基づきます:

- マスターテーブルのフィールドのみを使用する場合、リストボックス行にはマスターテーブルのセレクションの内容が表示されます。
- マスターテーブルに属さないフィールドを使用する場合、これらの"外部"テーブルはマスターテーブルにN対1でリレートしていなければなりません。そうでない場合"外部"フィールドは空で表示されます。自動リレーションがマスターテーブルセレクションの各レコードごとに実行され、リストボックスにはリレートしたフィールドの対応するデータが表示されます。
  マニュアルリレーションを使用する場合、リストボックスにデータを表示するためにはリレーションを有効化するプログラムを書く必要があります。

リストボックス定義の矛盾のため列が空で表示される場合、アプリケーションモードで不正な列ごとにエラーメッセージが表示されます。

異なるケースを説明するために例題を使用します。

2つのテーブル[Companies]と[Employees]を持つデータベースがあります。

- [Companies]テーブルのカレントセレクションは以下の通りです:

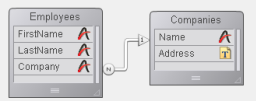
| **Company Name** |
|---|
| Big Encyclopedias |
| Tiny Computers |
| Boring Travel Company |

- [Employees]テーブルのカレントセレクションは以下の通りです:

| **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 |

[Companies]Nameフィールドがリストボックスの最初の列に関連付けられます。[Employees]First Nameと[Employees]Last Nameフィールドが続く列に割り当てられます。リストボックスのデータソースはカレントセレクションです。

- **ケース1**: 2つのテーブルが自動リレートされている場合




1) リストボックスのマスターテーブルが[Employees]の場合。リストボックスには[Employees]テーブルのカレントセレクションが表示され、各従業員の勤務先を表示するために自動リレーションが実行されます:

| 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) リストボックスのマスターテーブルが[Companies]の場合。リストボックスには[Companies]テーブルのカレントセレクションが表示されます。このセレクションには3レコードしかないため、リストボックスには3行表示されます。[Employees]First Nameと[Employees]Last Nameフィールドの列は空になります:

| Company Name | First Name | Last Name |
|---|---|---|
| Big Encyclopedias | | |
| Tiny Computers | | |
| Boring Travel Company | | |

- **ケース2**: 2つのテーブルはリレートされていない場合 (またはマニュアルリレートが使用されている場合)。

1) リストボックスのマスターテーブルが[Employees]の場合。リストボックスには[Employees]テーブルのカレントセレクションが表示されます。[Companies]Nameフィールド列は空です:

| 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) リストボックスのマスターテーブルが[Companies]の場合。リストボックスには[Companies]テーブルのカレントセレクションが表示されます。このセレクションには3レコードしかないため、リストボックスには3行表示されます。[Employees]First Nameと[Employees]Last Nameフィールドの列は空になります:

| Company Name | First Name | Last Name |
|---|---|---|
| Big Encyclopedias | | |
| Tiny Computers | | |
| Boring Travel Company | | |

他のテーブルのセレクションはプログラムで管理することができ、そうすればマスターテーブルに属さないフィールドが関連付けられた列を表示することができます。

# リストボックスにSQLクエリの結果を表示する

**リストボックスにSQLクエリの結果を表示する**

SQLクエリーの結果を直接配列タイプのリストボックスに表示することができます。これによりSQLクエリーの結果を素早く見る方法が提供されます。SELECTタイプのクエリーのみを使用できます。このメカニズムは外部SQLデータベースには使用できません。

この機能は以下の原則に基づいて動作します:

- クエリーの結果を受け取るリストボックスを作成します。リストボックスのデータソースは**配列**に設定しなければなりません。
- SELECTタイプのSQLクエリーを実行し、結果をリストボックスに割り当てた変数に受け取ります。Begin SQL/End SQL キーワードを使用できます (4Dランゲージリファレンス参照)。
- ユーザーはリストボックス列をソートしたり、更新したりできます。
- SELECTクエリーを実行するたびに、リストボックス列はリセットされます (複数のSELECTクエリーを実行して、リストボックスに行を追加することはできません)。
- SQLクエリーの結果受け取る列数と同じ数の列をリストボックスに用意することを推奨します。SELECTクエリーにより返される列数よりもリストボックスの列数が少ない場合、自動で列が追加されます。SELECTクエリーの結果よりも多い列数がある場合、不必要な列は自動で隠されます。
  **注:** 自動で追加された列には一時的な配列がバインドされます。この一時的な配列はフォームが閉じられるまで存在します。また一時的な変数が各ヘッダー用に作成されます。LISTBOX GET ARRAYSコマンドが呼び出されると、*arrColVars*引数には一時的な配列へのポインターが、*arrHeaderVars*引数には一時的なヘッダー変数へのポインターが返されます。例えば5つの列が追加されると、5番目の列の配列名は*sql_column5*でヘッダー名は*sql_header5*となります。
- インタープリターモードでは、SQLクエリーにより返されたデータに基づき、自動でリストボックスの既存の配列が型変更される場合があります。

**例**
PEOPLEテーブルのすべてのフィールドのデータを取得し、*vlistbox*という変数名のリストボックスに結果を表示します。これを行うためのメソッドは以下の通りです:

```
Begin SQL
   SELECT * FROM PEOPLE INTO <<vlistbox>>
End SQL
```

## サブフォームとウィジェット


概要

リストサブフォーム

ページサブフォーム

## 概要

サブフォームとは他のフォームに組み込まれるフォームのことです。

サブフォームには主に2つのタイプがあります:

- リストサブフォーム: 通常他のテーブルのデータを入力、表示、および変更するために使用されます。
- ページサブフォーム: 4Dウィジェットのような洗練されたインターフェースを作成するために使用します。

サブフォームは各タイプごとに特有のメカニズムやプロパティがあります。

### 用語

サブフォームに実装されたコンセプトを明確に説明するために、いくつかの用語についてここで定義します:

- **サブフォーム**: 他のフォームに組み込まれることを意図したフォーム。
- **親フォーム**: 1つ以上のサブフォームを含むフォーム
- **サブフォームコンテナ**: 親フォームに組み込まれた、サブフォームのインスタンスを表示するオブジェクト
- **サブフォームインスタンス**: 親フォームに表示されたサブフォームの実体。このコンセプトはとても重要です。親フォームには同じフォームを複数、サブフォームとして表示することができるからです。
- **リストフォーム**: データをリストとして表示するサブフォームインスタンス。
- **詳細フォーム**: リストサブフォームをダブルクリックすることでアクセスすることができる、ページタイプのサブフォーム。

### サブフォームの作成と設定

フォームにサブフォームを追加するには3つの方法があります:

- フォームウィザードで詳細オプションのサブフォームページを使用します (フォームウィザードを使用してフォームを作成する参照)。
- フォームエディターでオブジェクトバーのサブフォームツール を使用します。
- フォームエディターでエクスプローラーのフォームページからドラッグ＆ドロップします。

**注:** 定義済みオブジェクトライブラリーからウィジェットオブジェクトを追加すると、サブフォームタイプオブジェクト (ページフォーム) が作成されます (設定済みライブラリを使用する参照)。

必要に応じてこれらの方法を使い分けていただけます。

プロパティリストの"サブフォーム"テーマ内で**出力サブフォーム**オプションを選択したり選択解除したりすることで、フォームタイプを設定できます。この設定に基づき、選択されたサブフォームタイプに対応するオプションが表示されるよう、プロパティリストの他のオプションが更新されます。

### ウィジェット

4Dウィジェットは定義済みの複合オブジェクトです。プログラミングなしでもありでも利用できるこれらのウィジェットを使用すると標準の機能を簡単に実装できます。以下のウィジェットが提供されています:

- SearchPicker: 標準のアピアランスを持つ検索エリア
- DatePicker: 日付セレクター
- TimePicker: 時間セレクター

4Dウィジェットはページタイプのサブフォームで、特定の機能を実現するために作成されています。サブフォーム作成ツールまたは4Dに統合されたオブジェクトライブラリーからこれらのウィジェットを追加できます (設定済みライブラリを使用する参照)。

ウィジェットの利用方法は以下のマニュアルを参照してください: 4D Widgets（ウィジェット）

## リストサブフォーム

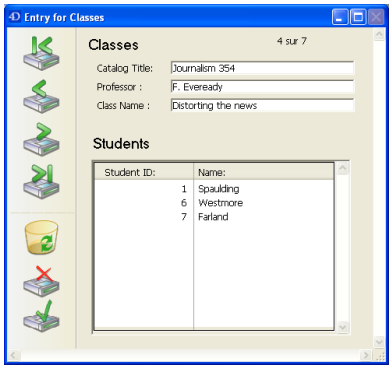
サブフォームは詳細フォーム中に表示される、他のテーブルに属するフォームです。リストサブフォームでは他のテーブルのデータを入力、表示、および更新できます。通常は1対Nリレーションが設定されたデータベースでリストサブフォームを使用します。

リレートされた1テーブルのフォーム上のリストサブフォームでは、リレートしたNテーブルのデータを入力、表示、および更新します。ひとつのフォーム上に、それぞれ異なるテーブルに属する複数のサブフォームを配置できます。しかしフォームの同じページに、同じテーブルに属するサブフォームを2つ以上配置することはできません。

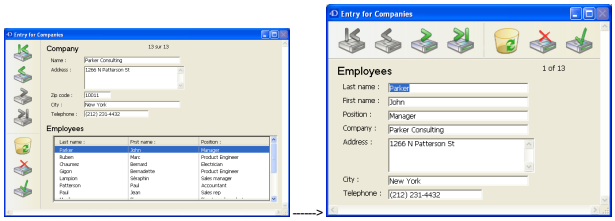
例えば連絡先管理データベースでは、ある連絡先のすべての電話番号を表示するためにリストサブフォームが使用されるでしょう。連絡先画面に電話番号が表示されますが、情報はリレートされたテーブルに格納されています。1対Nリレーションを使用することで、このデータベース設計は連絡先ごとに複数の電話番号を簡単に格納できるようになっています。自動リレーションにより、プログラムなしでリレートしたNテーブルへの直接データ入力がサポートされます。

通常リストサブフォームはNテーブルに結び付けられますが、それだけではなく、サブフォームインスタンスはすべての他のデータベーステーブルのレコードを表示できます。

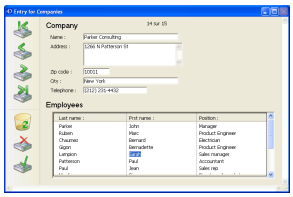
リストサブフォームをフォーム上に配置するには、フォームウィザードを使用するか、フォームエディター上で既存のフォームを追加します。サブフォームとして使用するリストフォームを作成しなければなりません。




リストサブフォームをデータ入力に使用するには2つの方法があります: ユーザーはサブフォームに直接データを入力できます。または詳細フォームを開いてデータを入力します。この設定ではサブフォームとして使用されるフォームがリストフォーム、入力のために使用されるフォームが詳細フォームとなります。




またリストフォームに入力するようにもできます:




### リストサブフォームの作成

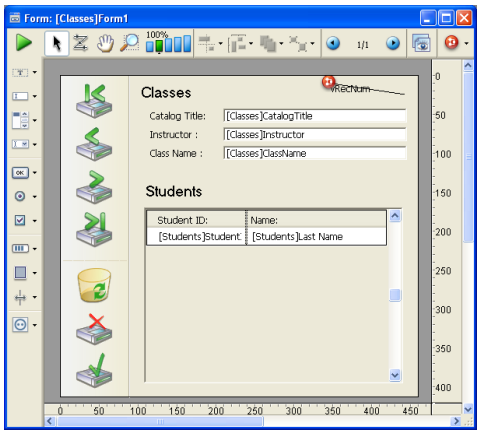
リストフォームを定義するには、まず親フォームで使用するサブフォームを作成しなければなりません。リストフォームに関する詳細は一覧フォームとレポートを参照してください。

親フォーム中では、フォームエディターのオブジェクトバーにあるサブフォームツールを使用してサブフォームを作成します 。

**注:** エクスプローラーのフォームページからドラッグ＆ドロップすることもできます。

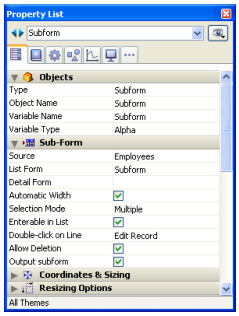
プロパティリストのサブフォームテーマで、**出力サブフォーム**オプションがチェックされていることを確認し、**ソース**メニューからソーステーブルを選択したら**リストフォーム**から使用するフォームを選択します。

親フォーム内でサブフォームコンテナをサイズ変更し、位置を調整することができます。




### リストサブフォームのプロパティ

リストサブフォーム特有のプロパティを設定できます。このプロパティにより、ユーザーアクションに許可するさまざまな自動機能を制御できます。




#### 変数名と変数型

通常これらのプロパティはページサブフォームで使用されます。ページサブフォームを参照してください。

#### ソース

このプロパティを使用してサブフォームのソースタイプを指定できます。リストサブフォームの場合、サブフォームが属するテーブルを選択します。

#### 詳細フォーム

このオプションを使用して、リストサブフォームに詳細フォームを割り当てます。詳細フォームはサブレコードを入力したり表示したりするために使用します。通常詳細フォームにはリストサブフォームより多くの情報が含まれています。詳細フォームはサブフォームと同じテーブルに属していなければなりません。詳細フォームを指定しない場合、4Dは自動でデフォルト入力フォームを使用します。

サブフォームの設定に基づき、ユーザーがサブレコード上でダブルクリックするか、サブレコードを追加/編集するコマンドを追加すると、詳細フォームが表示されます。

**注:** リストサブフォームに詳細フォームを結び付けるには、エクスプローラーから詳細フォームをサブフォームコンテナに、Shiftキーを押しながらドラッグすることもできます。

#### 自動幅

他のフォームオブジェクトと同様、サブフォームエリアも自由にサイズ変更できます。
またすべてのフィールドが表示されるよう、自動でサブフォームの幅を決定することもできます。これを行うにはオブジェクトコンテナプロパティで**自動幅**オプションを選択します。

**注:** エクスプローラーからフォームにドラッグでサブフォームを追加した場合、4Dは自動で適切な幅をサブフォームに設定します。

#### 選択モード

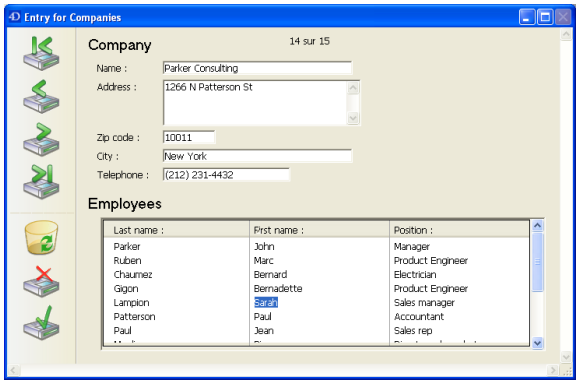
リストサブフォームでは3つの選択モード**なし**、**単一**、**複数**があります

- **なし**
  このモードが選択されているとレコードを選択することはできません。**リスト更新可**オプションが選択されていない限り、リスト中でのクリックは何の効果もありません。ナビゲーションキーはリストをスクロールするだけです。On Selection Changeフォームイベントは生成されません。
- **単一**
  このモードでは一度に一つのレコードのみを選択できます。レコードをクリックするとそれが選択され、カレントレコードになります。レコード上で**Ctrl+クリック** (Windows) や **Command+クリック** (Mac OS) 行うと選択されたり、選択が解除されたりします。
  **上**や**下** 矢印キーを押すと、リスト中の前後のレコードが選択されます。選択されるレコードが変わるたびに、On Selection Changeフォームイベントが生成されます。
- **複数**
  このモードでは同時に複数のレコードを選択できます。選択されたサブレコードはGET HIGHLIGHTED RECORDSで取得できます。レコードをクリックするとそのレコードが選択されますが、カレントレコードは変更されません。レコード上で**Ctrl+クリック** (Windows) または **Command+クリック** (Mac OS) すると選択または選択解除されます。**Ctrl+クリック** (Windows) や**Command+クリック** (Mac OS) そして**Shift+クリック**キーにより複数選択できます。**上**や**下** 矢印キーを押すと、リスト中の前後のレコードが選択されます。選択されるレコードが変わるたびに、On Selection Changeフォームイベントが生成されます。

#### リスト更新可

リストサブフォームに**リスト更新可**プロパティが設定されていると、ユーザーはリスト中で直接レコードデータを更新できます。これを行うには、更新するフィールド上で2回クリックを行い、編集モードにします (ダブルクリックにならないようクリックの間隔を空けなければなりません)。

デフォルトでこのモードはすべてのリストサブフォームに対し有効です:




このオプションがチェックされていない場合、データ入力は詳細フォームを通して行わなければなりません。

#### 行をダブルクリック時のアクション

ユーザーのダブルクリック時にリストサブフォームがどのようにふるまうべきか、設定できます。以前の4Dバージョンで作成されたデータベースの場合、空行をダブルクリックした際の動作も設定できます (互換オプション)。

- **行をダブルクリック**: サブフォームレコード上でダブルクリックされた際のアクション。以下の選択肢があります:
  - **なにもしない**: ダブルクリックを無視します。
  - **レコード編集**: サブフォームレコード編集モードにします。"リスト更新可"オプションがチェックされている場合、直接リスト中で編集できます。そうでない場合、サブフォームに割り当てられた詳細フォームを通して編集します。
  - **レコード表示**: サブフォームに割り当てられた詳細フォーム上にレコードを表示します (読み込みのみ)。
- **空行をダブルクリック**: サブフォーム中の空行をダブルクリックした場合に実行されるアクション。以下の選択肢があります:
  - **なにもしない**: ダブルクリックを無視します。
  - **レコード追加**: サブフォーム中に新しいレコードを作成し、編集モードにします。"リスト更新可"オプションが選択されている場合レコードは直接リスト中に作成されます。選択されていない場合、レコードはサブフォームに割り当てられた詳細フォーム上に作成されます。

#### 削除を許可

デフォルトでユーザーは、**Del**や**Backspace**を使用してリストサブフォーム中のサブレコードを削除できます。

特定のアプリケーションではこの振る舞いがインターフェースの標準操作を妨げることがあるので、**削除を許可**オプションを使用してこの動作を無効にできます。

このオプションにチェックが入れられていないと、ユーザーはキーボードの削除キーを使用したサブレコードの削除ができなくなります。

#### フォーカス表示

サブフォームインスタンスは**フォーカス表示**プロパティ ("入力"テーマ) を持つことができます。サブフォームインスタンスがフォーカスを得ると、ユーザーはナビゲーションキーや (複数行選択可の場合) **すべてを選択**メニューコマンド等を使用してそれを制御できます。

サブフォームがフォーカスを得たり失ったりすると、親フォームでOn getting focusやOn losing focusイベントが生成されます。この場合、OBJECT Get pointerコマンド (またはFocus objectコマンド) はサブフォームのテーブルへのポインターを返します。

すべてのフォーカス可オブジェクトのように、**フォーカスの四角を隠す**プロパティを使用して、フォーカスの表示を無効にできます。

#### 印刷

サブフォームに表示しきれないレコードが存在する可能性があるため、サブフォームレコードの印刷を扱うために、4Dは"印刷"テーマの3つのオプションを提供します:

- **可変** (デフォルト)
  このオプションが選択されていると、4Dはすべてのサブレコードを印刷するためにサブフォームエリアを伸縮させます。
- **固定 (切り捨て)**
  このオプションが選択されていると、4Dはサブフォームエリアに表示されているサブレコードのみを印刷します。フォームは一度のみ印刷され、印刷されないレコードは無視されます。
- **固定 (全レコード)**
  このオプションが選択されていると、サブフォームエリアのサイズは保持されますが、4Dはすべてのサブレコードが印刷されるまでフォームを複数回印刷します。
  **注:**
  - サブフォームオブジェクトコンテナの両側にオブジェクトを配置することはできません。コンテナの両側におかれたオブジェクトはサブフォームの行ごとに繰り返されます。
  - 同じフォームページ上に同じテーブルのサブフォームインスタンスを複数配置して印刷することはできません。
  - 可変サイズのオブジェクトの印刷については可変サイズエリアの印刷を参照してください。

**サブレコード管理用のボタンを追加する**

サブフォームへのデータ入力を処理するカスタムボタンを追加できます。標準やハイライト、非表示などどのようなタイプのボタンでも使用できます。プログラムや (データ入力参照) 標準アクションを使用してアクションを設定できます。

4Dはサブレコードを管理する基本的なニーズに応える3つの標準アクション、**サブレコード編集**、**サブレコード削除**、**サブレコード追加**を提供します。フォームに複数のサブフォームインスタンスが含まれる場合、フォーカスを持つサブフォームにアクションが適用されます。

ボタンの追加や標準アクションの割り当てについてはボタンや標準アクションを参照してください。

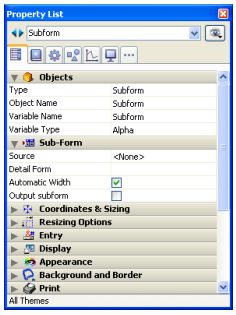
# ページサブフォーム

ページサブフォームはカレントサブレコードのデータやコンテキストに基づく関連する値 (変数やピクチャーなど) を表示できます。ページサブフォームを使用する利点の一つは、上級機能を使用できるようになることであり、親フォームと直接に相互作用を行うことができるようになります。ページサブフォームには特別なプロパティやイベントがあり、プログラムから完全に制御することができます。

**注:** サブフォームを使用して追加の機能を提供するコンポーネントを生成できます。詳細は4Dコンポーネントの開発とインストールを参照してください。

## ページサブフォームの作成

サブフォームのページモードを有効にするには、プロパティリストの出力サブフォームのオプションのチェックを外します。この場合、リストサブフォームの設定に関連するプロパティ (選択モード、ダブルクリック時の動作等) は表示されなくなります:




ページサブフォームは"詳細フォーム"プロパティで指定されたフォームを使用します。リストサブフォームと異なり、使用されるフォームは親フォームと同じテーブルでもかまいません。またプロジェクトフォームを使用することもできます。実行時、ページサブフォームは入力フォームと同じ標準の表示特性を持ちます。リストフォームのメカニズム (マーカーの管理など) は無効になります。

## ページサブフォームのプロパティ

親フォームへの統合と相互作用を容易にする特別なプロパティを使用してページサブフォームを管理します。

### 変数名

サブフォームオブジェクトに変数をバインドできます。デフォルトでこの変数名は"Subform"となっています。変数には型があり (後述) 親フォーム内で標準変数として表示できます。この変数を変更するとフォームイベントが生成され、親フォームとサブフォームの値を同期することができます:

- On Data Changeフォームイベントを使用して、サブフォーム内で変数値が変更されたことをサブフォームコンテナに値が変更されたことを通知できます。
- On bound variable changeフォームイベントを使用して、親フォーム内で変数値が変更されたことをサブフォーム (サブフォームのフォームメソッド) に通知します。

### 変数タイプ

このプロパティを使用して、サブフォームオブジェクトにバインドされる変数の型を指定できます。デフォルトで文字型に設定されています。変数タイププロパティは、この変数を通して親フォームとサブフォームの間で交換される値の性質を決定します。
タイプを"なし"に設定することもできます。この場合4Dは実行時に自動で型を決定しますが、コンパイル時に型指定エラーとなる点に留意してください (ランゲージリファレンスの変数、"ダイナミック変数"参照)。

### ソース

このプロパティでは異なるタイプのソースを選択できます:

- ***<なし>***: サブフォームとしてプロジェクトフォームやコンポーネントフォームを使用する場合に選択します。これらのサブフォームはページサブフォームでのみ利用できます。"出力サブフォーム"オプションが選択されているとこれらのフォームは使用できません。
コンポーネント内で公開されているコンポーネントフォームは"詳細フォーム"リストに表示されます (サブフォームを公開 (コンポーネント)参照)。
- ***テーブル名***: テーブルフォームを使用するためにはこのオプションを選択します。

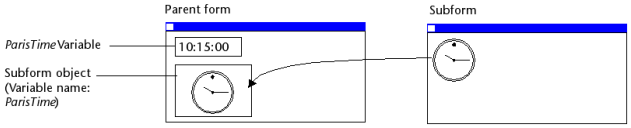
### オブジェクトメソッドとイベント

サブフォームオブジェクトにオブジェクトメソッドを記述し、(生成されるイベントに基づき) その機能を制御することができます。ページサブフォームにより管理されるイベントに関する説明はForm eventコマンドの説明を参照してください。

特別なメカニズムを使用して、サブフォームオブジェクトの内容と親フォームの内容との相互作用を管理しなければならない点に留意してください (上級フォーム間通信プログラム参照)。

## バインドされた変数の管理

サブフォームにバインドされた変数を使用して2つのコンテキスト (フォームとサブフォーム) をリンクし、洗練されたインターフェースを作成することができます。例えばサブフォームに時計を置くとします。サブフォームが置かれる親フォームには時間型の入力可変数が置かれています:



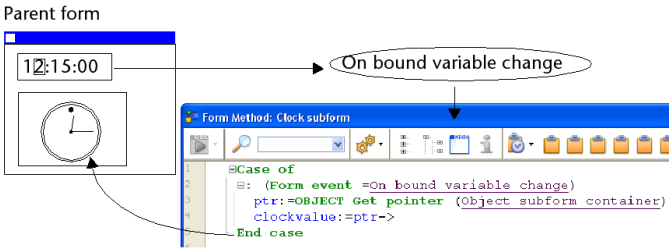

両オブジェクト (時間変数とサブフォームコンテナ) には同じ変数名が割り当てられています。この場合、親フォームを開いたとき、4Dは自動で両方の値を同期化します。変数の値が複数の場所で設定されている場合、4Dは最後にロードされた値を使用します。以下のロード順が適用されます:
1-サブフォームのオブジェクトメソッド
2-サブフォームのフォームメソッド
3-親フォームのオブジェクトメソッド
4-親フォームのフォームメソッド

親フォームが実行されるとき、開発者は適切なフォームイベントを使用して変数ので同期を処理しなければなりません。2タイプの相互作用、親フォーム->サブフォーム、またはサブフォーム->親フォームが発生します。

### サブフォームの内容を更新

ケース1: 親フォームの変数の値が更新され、この更新をサブフォームに反映させる必要があります。この例では*ParisTime*変数の値がユーザー入力、あるいは動的に (例えばCurrent timeコマンドで) 12:15:00に変更されています。

この場合On bound variable changeフォームイベントを使用します。このイベントはサブフォームプロパティで選択されなければなりません。このイベントはサブフォームのフォームメソッドで生成されます。




On bound variable changeフォームイベントは以下のときに生成されます:

- 親フォームの変数に値が割り当てられたとき (同じ値が代入された場合でも) で、
- サブフォームが0ページまたはカレントフォームページにおかれているとき。

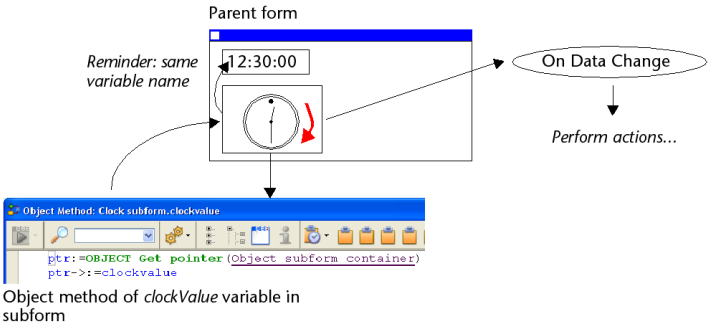
先の例のとおり、直接変数名ではなく、サブフォームコンテナへのポインターを返す OBJECT Get pointer コマンドの利用を推奨します。親フォームに同じサブフォームを複数配置することが可能だからです (例えば複数の時間帯を表示ために複数の時計を表示するウィンドウ)。この場合ポインターを使用することでのみ、どのサブフォームコンテナがイベントの生成元かを知ることができます。

#### 親フォームの内容を更新

ケース2: サブフォームの内容が更新され、その更新を親フォームに反映させる必要があります。この例ではユーザーが時計の針を手動で動かすことができるようなケースです。

この場合、サブフォームから、親サブフォームコンテナの変数にオブジェクトの値を割り当てなければなりません。先の例の通り、OBJECT Get pointerコマンドをObject subform containerセレクターで使用し、サブフォームコンテナのポインターを得ることをお勧めします。

変数に値を割り当てると親サブフォームコンテナのオブジェクトメソッドでOn Data Changeフォームイベントが生成され、メソッドによるアクションを実行できます。このイベントはサブフォームコンテナのプロパティリストで選択されていなければなりません。




Object method of *clockValue* variable in subform

**注:** 時計の針を手動で動かすと、サブフォーム中の*clockValue*変数のオブジェクトメソッドでOn Data Changeフォームイベントが生成されます。

#### 上級フォーム間通信プログラム

親フォームとサブフォームインスタンス間の通信では、バインドした変数を通して値を交換する以上のことを行うこともできます。実際親フォームで行われたアクションに基づきサブフォーム中の変数を更新したり、またはその逆を行いたい場合があるでしょう。先の"動的な時計"タイプのサブフォームの例で言えば、各時計ごとにアラーム時刻を複数設定したい場合があります。

このようなニーズにこたえるため、4Dは以下のメカニズムを実装しています:

- OBJECT Get nameコマンドで"subform"引数を使用してサブフォームオブジェクトを指定し、OBJECT Get pointerコマンドを使用する、
- CALL SUBFORM CONTAINERコマンドを使用してサブフォームからコンテナオブジェクトを呼び出す、
- EXECUTE METHOD IN SUBFORMコマンドを使用してサブフォームのコンテキストでメソッドを実行する

##### Object get pointerとObject get nameコマンド

Object subform containerセレクターに加え、OBJECT Get pointerコマンドは第二引数に指定した名前を持つオブジェクトを検索する際に、どのサブフォーム内を検索するか指定するための引数を受け入れます。このシンタックスはObject namedセレクターが渡された場合のみ使用できます。

例えば以下の式は:

```
$ptr:=OBJECT Get pointer(Object named;"MyButton";"MySubForm")
```

"MySubForm"サブフォームオブジェクト中の"MyButton"オブジェクトに割り当てられた変数へのポインターを返します。このシンタックスを使用すれば親フォームからサブフォーム内のオブジェクトにアクセスできます。
またOBJECT Get nameコマンドを使用すればフォーカスを持つオブジェクトの名前を取得できます。

##### CALL SUBFORM CONTAINERコマンド

CALL SUBFORM CONTAINERコマンドを使用すると、サブフォームインスタンスからサブフォームコンテナオブジェクトにイベントを送信できます。その結果親フォームのコンテキストで処理が可能となります。イベントはコンテナオブジェクトメソッドで受信されます。(クリックやドラッグ＆ドロップなど) サブフォームにより検知されたすべてのイベントの発生元となりえます。

送信するイベントコードに制限はありません (例えば20000や-100など)。既存のイベントに対応するコード (例えばOn Validateに対応する3) を使用することも、カスタムコードを使用することもできます。前者のケースでは、サブフォームコンテナのプロパティリストでチェックを入れたイベントのみを使用できます。後者の場合、使用するコードは既存のフォームイベントに対応してはいけません。将来の4Dバージョンで番号が衝突しないようにするために、負数を使用することをお勧めします。

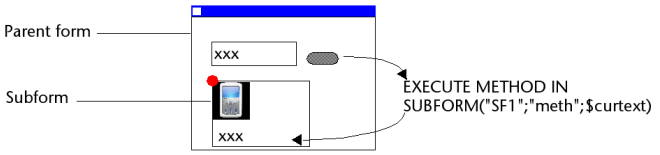
詳細はCALL SUBFORM CONTAINERコマンドの説明を参照してください。

##### EXECUTE METHOD IN SUBFORMコマンド

EXECUTE METHOD IN SUBFORMコマンドを使用するとフォームやオブジェクトから、(サブフォーム変数やオブジェクト等にアクセスできる) サブフォームインスタンスのコンテキストでメソッドの実行をリクエストできます。このメソッドは引数を受け取れます。

このメカニズムを図示すると以下のようになります:

*Example of execution of EXECUTE METHOD IN SUBFORM command*




詳細はEXECUTE METHOD IN SUBFORMコマンドの説明を参照してください。

##### GOTO OBJECTコマンド

GOTO OBJECTはサブフォームから実行されても、親フォームオブジェクトから目的のオブジェクトを検索します。

# 一覧フォームとレポート


概要

一覧フォーム

レポート印刷用のフォーム

出力コントロールラインを使用する

レポートの例題

可変サイズエリアの印刷

差し込み印刷ドキュメントを作成する

ラベルの作成

## 概要

一覧フォームはスクリーン上のレコードを一覧表示したり、レポートを印刷する目的で使用します。

クイックレポートを使用すると特定の用途で使用できるレポートを素早く作成できます。しかしフォームエディターを使用すればよりカスタマイズされたレポートを作成できます。詳細はクイックレポートを参照してください。

**注:** 一覧フォームとして使用できるのはテーブルフォームのみです。プロジェクトフォームはページモードでのみ使用できます。

### 一覧フォームエリア

一覧フォームは異なるプロパティを持つ複数のエリアで構成されます:

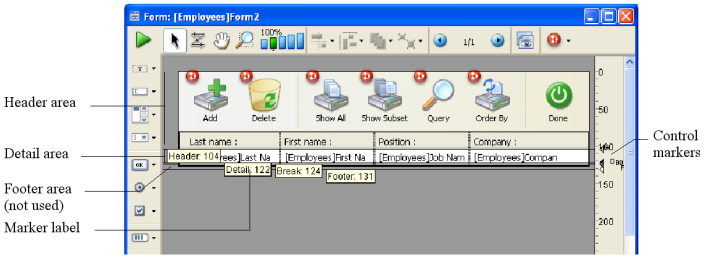
- **ヘッダーエリア**: レポートのタイトル、列ヘッダー、そしてフォームを管理するボタンやオブジェクトを配置します。
- **詳細エリア**: レポートのボディー (中身) を配置します。
- **フッターエリア**: ボタンや、レポート中の全レコード情報に基づくサマリなどを配置します。
- **ブレークエリア**: レコードリストの下に表示されるテキストやグラフ、そして全レコードやレコードのサブグループに基づくサマリなどを配置します。

フォームウィザードを使用して一覧フォームを作成すると、これらのエリアが自動で作成されます。フォームタイトル (テーブル名) やフィールド名および制御ボタンがヘッダーエリアに配置されます (実際に配置される内容はフォームウィザードの設定状況により変わります)。選択したフィールドは詳細エリアに配置されます。ブレークエリアも作成されますが、フォームウィザードはそこに何も配置しません。

フォームエディターでフォームを開いたら、これらの各エリアのサイズや配置するオブジェクトを変更したり、ブレークエリアに新しくオブジェクトを追加したりできます。また新しくブレークエリアを作成してサマリを追加することもできます。

ヘッダー、詳細、ブレーク、フッターエリアとして動作するフォームのエリアはマーカーラインを使用して制御されます。マーカーラインを縦にドラッグして各エリアのサイズを変更できます。

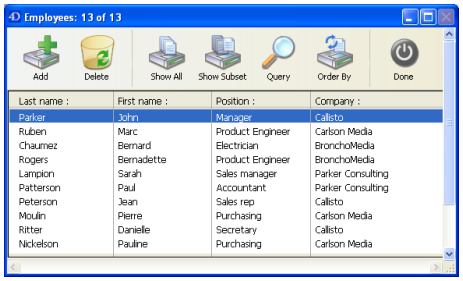
以下の図はフォームウィザードのXPテンプレートを使用して作成した、基本的な一覧フォームです:




**注:** マーカーラインやそのラベルの表示/非表示を切り替えることができます (フォームエディター参照)。

横に引かれた線を使用してヘッダー、詳細、ブレーク、フッターエリアに分けられます。フォームの一番上からヘッダー線までがヘッダーエリアです。同様にヘッダー線から詳細線までの間が詳細エリア、そしてフッターエリアは (B0) ブレーク線からフッター線までの間です。各マーカーラベルをドラッグすれば各エリアのサイズを変更できます。

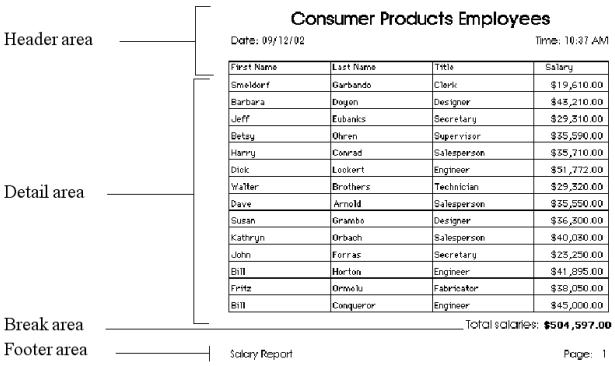
このフォームがアプリケーションモードで実行されると以下のように表示されます:




詳細エリアはフォームウィンドウのサイズに応じて伸縮します。ヘッダーやフッターは固定サイズです。
スクリーンに一覧フォームが表示される場合、ヘッダーとフッターにはクリック可能で入力不可のオブジェクト (ボタンやラジオボタン、階層リスト等) を配置できます。詳細は後述のマーカーラインに関する記述を参照してください。

印刷されるレポートの場合、ヘッダーには通常日付や時刻、レポート名や列タイトルが配置されます。レコードは詳細エリアに表示されます。計算値はブレークエリアに配置し、フッターにはページ番号が配置されます。

印刷されるレポートの例:




ブレークレベルを追加して小計や他の計算値を印刷することもできます。またレポートのボディ部に表示されるヘッダーを追加することもできます。追加のヘッダーエリアはサブグループを識別するために使用されます。複数のヘッダーやブレークエリアを使用する例については出力コントロールラインを使用するを参照してください。

# 一覧フォーム

### 標準機能

一覧フォームにはレコードリストが表示されます。どんなフォームでも出力フォームとして使用できますが、一覧フォームは通常以下の機能を備えています:

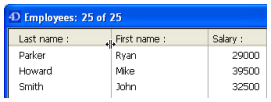
- 各行にレコードが表示されます。
- 各列にはフィールドや変数が表示されます。
- 各列のラベルはウィンドウ上部に表示されます。各タイトルエリアの間にあるスプリッターを使用して列のサイズを変更できます。
- ヘッダーやフッターにはボタンやポップアップメニューを配置できます。

#### リストのスクロール

一覧フォームには標準の横および縦スクロールバーが表示されます。スクロールバーを操作することでレコードをスクロールできます。また**Page Up**や**Page Down**キーを使用してページ単位でスクロールしたり、**Home**や**End**キーでリストの先頭や最後に移動することもできます。
フィールドが一覧フォームのウィンドウに収まらない場合、ウィンドウ下部にある横スクロールバーを使用してスクロールできます。

#### 列のリサイズ

一覧フォームの列幅を変更することができます (この機能をフォームエディターで無効にしていない限り)。これを実行するにはマウスカーソルを列タイトルの間に合わせます。カーソルの形状が変化し、列幅を変更可能である旨が通知されます。この状態でクリックしながらマウスを左右に移動します:




#### レコードの選択

レコードをクリックすると行がハイライトされ、レコードが選択されたことが示されます。レコードは一行、連続した複数行、あるいは連続しない複数行を選択できます。レコードを選択した後、それらのレコードをサブセットとして表示したり削除したりできます。

- 一件のレコードを選択するには、一覧フォーム上に表示されているレコードをクリックするか、**上**または**下矢印**キーを押します。選択されたレコードがハイライトされます。
- 連続した複数行を選択するには、先頭または最後のレコードを選択し、**Shift**キーを押しながら最後または先頭のレコードをクリックします。2つのクリックされたレコードに挟まれたレコードが選択されます。
- 連続しない複数行を選択するには、まず一件レコードを選択し、**Ctrl**キー (Windows) または**Command**キー (Mac OS) を押しながら、選択したいレコードをクリックします。クリックしたレコードが選択されますが、既に選択されたレコードの選択は解除されません。
- 一覧フォーム中に表示されているレコード (カレントセレクション) をすべて選択するには、**編集**メニューの**すべてを選択**を選択するか、**Ctrl+A** (Windows) または **Command+A** (Mac OS) ショートカットを使用します。

テーブル中の全レコードを表示して選択するには、まず**レコード**メニューから**すべてを表示**を選択し、そののちに**編集**メニューの**すべてを選択**を選択します。

#### レコードの追加と編集

一覧フォーム内で直接レコードを追加したり編集したりできます。一覧フォームでは複数のレコードが同時に表示されているので、連続してデータを更新する場合に特に便利です。

カレントテーブルのフィールドを入力したり編集したりできます。一覧フォームでは変数や他のテーブルのフィールド、サブフォームには入力/編集できません。

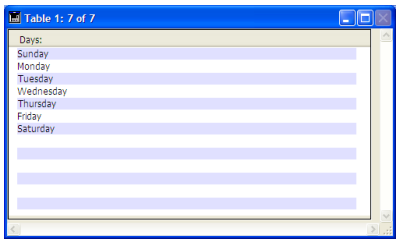
一覧フォームでは以下のことが可能です:

- フィールドへの入力 (入力モードにするにはフィールドを選択した後もう一度クリックします)
- **Tab**、**Enter** または **Return** キーを使用してフィールドからフィールド、レコードからレコードに移動
- **レコード** > **新規レコードをリストに追加**メニューを使用して新規レコードを追加

**Tab**、**Enter** または **Return** キーを押すか、他のフィールドをクリックすると変更が保存されます。

#### 空行の管理

一覧モードでは必要に応じて最後のレコードの後に空行が表示されることがあります:



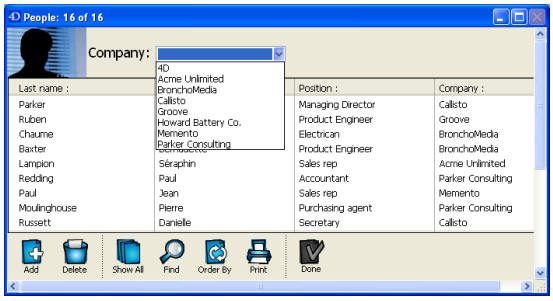

ウィンドウにフォームが表示される際、各空行ごとにOn Display Detailフォームイベントが生成されます。この場合カレントレコードはありません。Record numberコマンドは-1を返し、Selected record numberは意味のある値を返しません。このイベントでDisplayed line numberコマンドを使用すれば現在表示を行っている行番号を知ることができます。

### 一覧フォームのカスタマイズ

フォームウィザードで作成した一覧フォームをそのままでも使用できますが、必要であれば以下のようなカスタマイズを行うことができます:

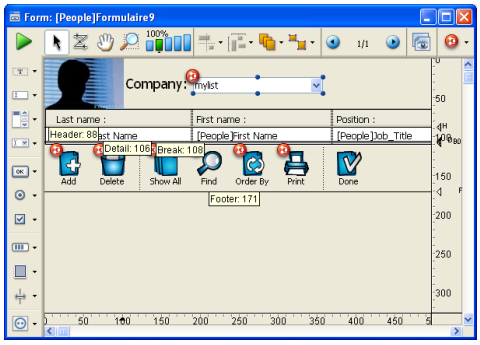
- フィールドを変数に置き換えたり、オブジェクトメソッドを記述したりする
- オブジェクトのフォントや枠線、カラー属性等を変更する
- 列の幅を変更する
- マーカーラインを移動して各エリアの高さを変更する
- ヘッダーエリアにロゴなどの画像を追加する
- ヘッダーやフッタいに追加のデータを表示する

以下の図は名刺を管理する、カスタマイズされた一覧フォームの例です:




制御ボタンがフッターエリアに移動され、ヘッダーにはピクチャーが追加されています。またヘッダーエリアにドロップダウンが配置され、会社で名刺を検索できるようになっています。

このフォームをフォームエディターで開いたのが以下の図です:




例えばヘッダーエリアに画像を配置するためにマーカーが移動され、ヘッダーの高さが変更されています。

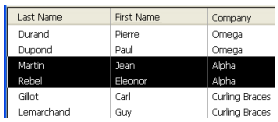
#### レコード毎に複数行を表示する

一覧フォームでフィールドを複数行に表示できます。詳細エリアの高さを調整し、必要なエリアを確保します。そしてオブジェクトを縦に配置します。一レコード内で複数行を表示する場合は、線や四角などのオブジェクトを使用してフィールドをグループ化するとよいでしょう。

以下は一レコードあたり二行を表示する一覧フォームの例です:

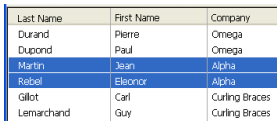
#### システムハイライトカラー

v2004より前の4Dでは選択されたレコードは反転色を使用して描画されていました。各ピクセルごとに色が反転し、例えば白が黒に、赤が緑になりました。この原則は背景色および文字の描画に適用されました:



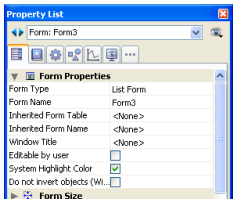
| Last Name | First Name | Company |
|---|---|---|
| Durand | Pierre | Omega |
| Dupond | Paul | Omega |
| Martin | Jean | Alpha |
| Rebel | Eleonor | Alpha |
| Gillot | Carl | Curling Braces |
| Lemarchand | Guy | Curling Braces |

v2004より、OSのインターフェース仕様に沿うために、4DはOSのシステム設定を使用するようになりました:



| 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 |

**システムハイライトカラーオプション**を使用すれば、変換されたデータベースでもこのインターフェースを使用することができます。このオプションは2004より前のデータベースから変換されたストラクチャーで、一覧フォームにのみ表示されます:




このオプションが選択されると、4Dは今までの反転色を無効にし、システムハイライトカラーを使用して選択されたレコードを表示するようになります。標準のインターフェース仕様に沿うために、このオプションを使用する際は適切なテキスト描画色および背景色を選択しなければなりません:

- 各オブジェクトのテキストカラー: **自動**オプション
- 各オブジェクトの背景色カラー: **透過**オプション
- フォームの詳細エリアにある四角のカラー: **自動**オプション

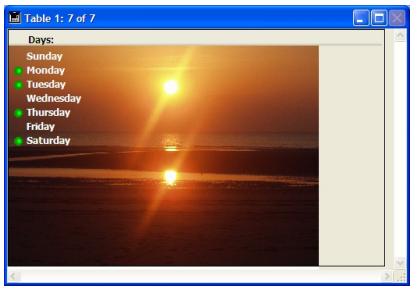
#### オブジェクトの表示オプション

一覧フォームではすべてのオブジェクトのプロパティで**表示可**プロパティを使用できます。このプロパティでは以下の3つから値を選択できます:

- 常に表示
- レコードが選択されている場合
- レコードが選択されていない場合

このプロパティは一覧フォームのボディ部に配置されたオブジェクトを描画するときにのみ使用されます。このプロパティは処理中のレコードが選択されているかいないかの状態に基づき、そのオブジェクトを描画するかしないかを決定するために使用されます。

このプロパティを使用すればカラー以外の方法でレコードの選択状態を表すことができます:




オブジェクトがOBJECT SET VISIBLEコマンドで非表示にされている場合、4Dはこの属性を考慮せず、選択状態に関わらず非表示のままとなります。

最後のレコードの下に表示される空行の場合、**常に表示**または**レコードが選択されていない場合**プロパティを持つオブジェクトが各空行に表示されます。

# レポート印刷用のフォーム

4Dではフォームまたはクイックレポートエディター (クイックレポート参照) を使用してレポートを生成できます。

フォームを使用してレポートを作成することにはいくつかの利点があります。フォーム上に画像を配置できる、精密にレポート要素の配置をコントロールできる、メソッドを使用して複雑な計算を行える、そしてブレークレポートの各サブセクションを識別するためにヘッダーを使用できるなどです。フォームレポートはユーザーフォームエディターを使用して各ユーザーがカスタマイズを行うことも可能です (ユーザーフォーム参照)。

以下のようなレポートを作成するためにフォームを使用できます:

- 列形式でないフォーマットを使用する場合
- サブフォームを使用する場合
- 埋め込み画像を使用する場合
- 極細線などの特別な画像要素を使用する場合

画面に表示されるフォームと異なり、印刷されるレポートではレポートの最後でブレークエリアを使用できます。各ページ毎に下部に表示されるオブジェクトをフッターエリアに置きます。

## レポートのタイプ

### 印刷用一覧レポート

情報を列に表示したリストは、電話番号、料金表、仕様書、部品リストなど、印刷レポートによく使用されます。
フォームウィザードを使用して一覧レポートを作成するにはフォームタイプとして**印刷用リストフォーム**を選択します。

画面にレコードを表示するリストフォームと同様に、印刷用リストフォームにも一連の情報が列に表示され、列タイトルを表示したり、画像を配置したりできます。レポートの最後に印刷されるブレークエリアには総計などの計算値を配置します。

総計を計算するためにメソッドを使用する方法については後述の例題を参照してください。

### ページレポート

一ページに一レコードを印刷するようなレポートもあります。例えばデータベースから請求書やレターなどを印刷するような場合です (差し込み印刷ドキュメントを作成する) 参照。

このようなレポートを作成する場合、フォームウィザードのフォームタイプで**印刷用詳細フォーム**を選択します。

ヘッダー (H) マーカーをページの一番上に配置し、フィールドやその他のレポート要素をその下に配置します。

レポートにサブフォームが含まれる場合、サブフォームエリアのサイズを固定にし、エリアに表示されるフォームのタイプを印刷に設定してください。そうするとサブフォームエリアはそのサイズで印刷され、次のページにまたがることはありません。固定長フレームの印刷については可変サイズエリアの印刷を参照してください。

詳細 (D)、ブレーク (B0)、そしてフッター (F) マーカーは用紙サイズの一番下に配置します。これにより一ページに一レコードしか印刷されなくなります。

### サブフォームを使用する

請求書の印刷はアプリケーションでよくある最も基本的な機能です。典型的な請求書には宛先名と請求先所在地、配送先所在地、請求金額があり、さらに購入物品のリストが続きます。

請求アプリケーションではページモードの印刷レポートフォームを作成します。このフォームには2つのテーブルからデータを取得し、印刷します。[Orders]テーブルからは顧客情報 (請求先所在地、配送先所在地) を、[Line Items]テーブルからは購入物品を取り出します。総請求額は計算で求められ、[Orders]テーブルに格納されます。

請求書用フォームは[Orders]テーブルフォームとして作成され、[Line Items]テーブルのリストフォームをサブフォームとして含みます。印刷時にサブフォームは、すべての購入物品を印刷するために拡張されるかもしれません。詳細は可変サイズエリアの印刷を参照してください。

詳細エリアはページいっぱいに取られます。各売上オーダーごとにひとつの請求書が印刷され、購入物品は必要なだけ印刷されます。

### テキストフィールド

多くのアプリケーションではテキストフィールドに可変長のメモや説明を入力することができます。テキストフィールドを詳細エリアで可変長オブジェクトとして印刷できます。
この場合テキストフィールドは印刷時に、その内容をすべて印刷するため縦方向に拡張されます。テキストフィールドの印刷に関する詳細は可変サイズエリアの印刷を参照してください。

### レターの宛名

郵便物のラベルを作成したい場合、特別な印刷用フォームをデザインできます。グラフィック要素やフォントの設定、変数の使用なども可能です。

**注:** ラベルを使用すると、素早く簡単にラベルを作成できます。

ラベル印刷用に出力フォームを作成する方法についてはPRINT LABELコマンドを使用するを参照してください。

## ブレークを使用する

ブレークやブレークヘッダーを使用したレポートを印刷することができます。ブレークはレコードのソートに基づいて作成されます。

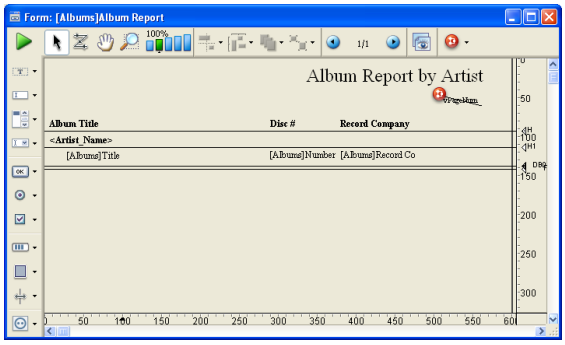
4Dデータベースで CDの楽曲リストを管理しているとしましょう。このリストをアーチストごと印刷するとします。アーチストでレコードをソートすると、すべてのレコードはいずれかのアーティストグループに属します。ここでブレークは印刷される各グループの最後のレコードの後に発生します。画面上でレポートをプレビューした結果は以下の通りです:

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は情報を魅力的に表示するために使用することのできる機能を提供しています。このフォームをエディターで開くと以下の通りです:




**注:** ブレークレベルやヘッダーを使用するレポートを構築するためには、まずブレーク処理を起動しなければなりません。ブレーク処理を起動するために使用することのできる方法については後述の"ブレーク処理を起動する"を参照してください。

ブレークヘッダーはレコードグループが印刷される前に一回印刷され、ブレークはレコードグループが印刷された後に一回印刷されます。先の例では並び替えに使用された最初のフィールドの結果としてブレークが発生するため、ブレークが"レベル1ブレーク"、ブレークヘッダーが"レベル1ブレークヘッダー"と呼ばれます。

最大9個のブレークを設定できます。ブレーク処理を起動するためにSubtotalを使用する場合、使用するブレーク数よりも1つ多い数のフィールドでソートしなければなりません。1ブレークを使用する場合、2つのフィールドでソートしなければなりません。3ブレークを使用する場合、4つのフィールドでソートしなければなりません。

レポートフォーム中で、ブレークレベルやブレークヘッダーレベルの追加は、エリアの追加に対応していなければなりません。これらのエリアはマーカーラインを追加することによって行います。マーカーラインの追加方法については出力コントロールラインを使用するを参照してください。

ここではブレークを使用するレポートの作成方法について説明します。完全な例題はレポートの例題を参照してください。

#### ブレーク処理を起動する

ブレークヘッダーエリアや累計小計値、その他のブレーク処理を印刷するためには、まずレポートフォーム上でブレーク処理を起動しなければなりません。ブレーク処理の起動は以下のいずれかの方法で行います:

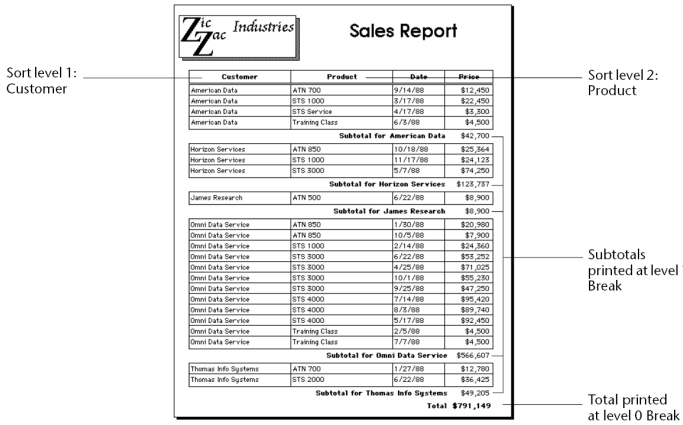
- フォームメソッドまたはオブジェクトメソッドにSubtotal関数を配置する
- レポート印刷開始前にBREAK LEVELとACCUMULATEコマンドを実行する

Subtotal関数を使用する場合、使用するブレークレベル数よりも1多いフィールド数でソートを行わなければなりません。例えば2ブレークレベルを使用する場合、3フィールドでソートします。

#### レポートで小計を使用する

このセクションではソートの順番がどのようにレポートに影響するのか、またブレークを追加して小計を作成する方法について説明します。

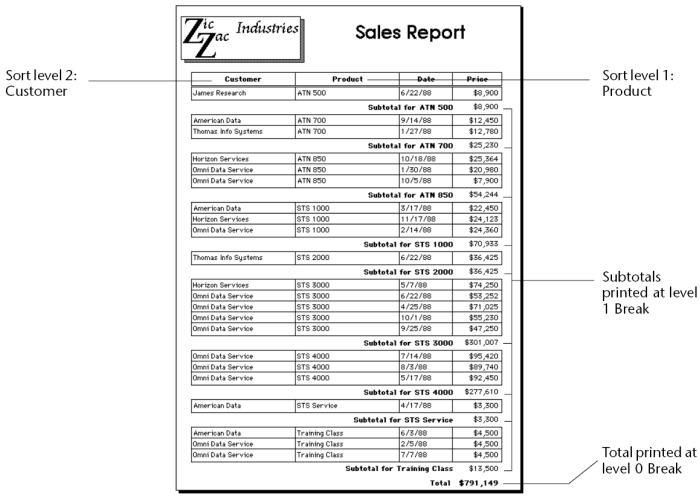
自動で総計や小計を計算させ、印刷することができます。以下のレポートでは各顧客ごとに小計を、レポートの最後では総計を計算しています。




これらのレコードは顧客および製品でソートされています。各顧客のレコードグループが印刷された後、その顧客の小計が計算され印刷されます。すべての顧客のレコードが印刷されたあと、総計が印刷され印刷されます。

小計の計算と印刷を行うタイミングは、最初のソートフィールドの値が変わったときであり、そこがブレークとなります。顧客ごとの小計はレベル1のブレークと呼ばれます。これは顧客フィールドが最初のソート条件フィールドであり、その値が変わったところでブレークが生成されるからです。総計はレベル0のブレークで計算されます。レベル0のブレークはすべてのレコードが印刷されたあと、レポートの最後に印刷されます。

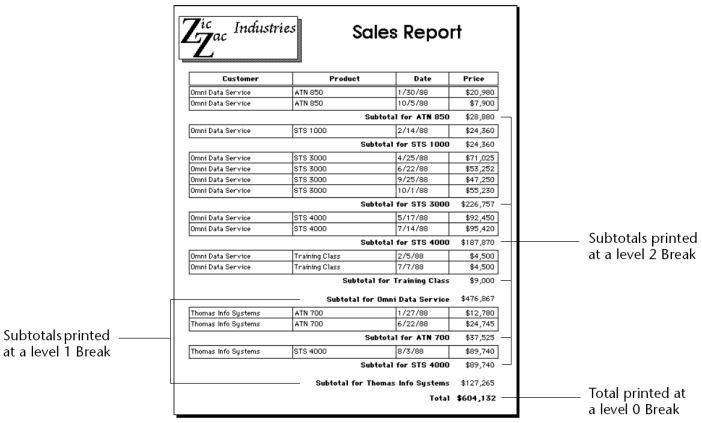
以下の図はブレーク時の印刷に関するもう一つの例題です。同じレコードを使用していますが、異なるブレークを生成するために異なるソート順を使用しています。今回はレコードを製品そして顧客でソートしています。小計は製品名が変わったときに計算されます。これは先ほどと同じレベル1のブレークですが、異なるフィールドに対して生成されています。




#### 追加のブレークレベル

ソートレベルとブレークレベルを追加することで、追加のサマリ計算を提供できます。

下の図は売り上げの記録を示していて、顧客/製品/営業担当でソートされています。サマリ計算では2セットの小計、各顧客ごとの小計と、さらに各顧客内で、製品ごとの小計が印刷されています。最後にこのレポートは会社全体の総計を印刷します。これはそれぞれレベル2ブレーク、レベル1ブレーク、そしてレベル0ブレークを実行する例題です。

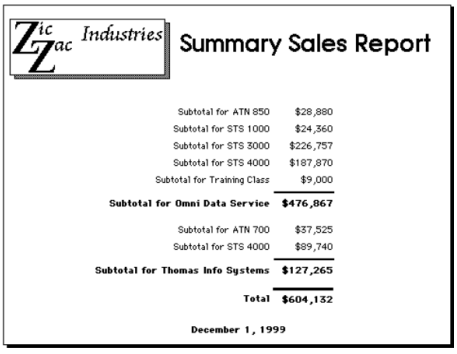



小計の計算はブレークを生成するレコードグループでのみ実行されます。例えばこの例では各顧客ごとに販売された各製品ごとに、そして各顧客に販売された製品すべての小計が計算されます。

**注:** ブレークレベルの追加については出力コントロールラインを使用するを参照してください。

**サマリレポート**

サマリ (概要) 情報だけを印刷するレポートを作成することもできます。このようなレポートでは小計と総計だけがそのタイトルとともに印刷されます。以下の図はサマリ情報のみを印刷するレポートです:

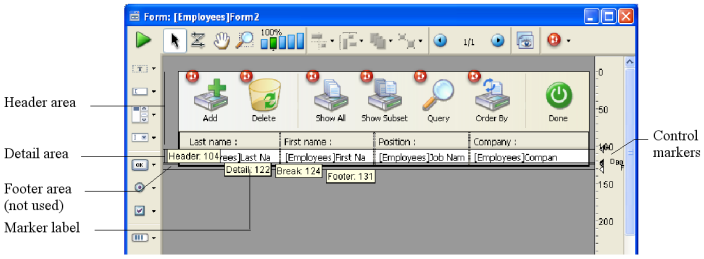



このレポートではレコードが顧客/製品/日付でソートされ、ソート順で作成されたブレークで計算が実行されます。各レコードの詳細エリアは印刷されません。レコードは計算のためにのみ使用されます。この場合でも4Dはレポートの印刷中、計算のためにレコードを先頭から最後までロードすることに留意してください。このような種類のレポートは詳細マーカーをヘッダーマーカーと同じ位置に置き、詳細エリアの高さを0にすることで実現します。

小計、平均、最大、最小値を追加で計算し、フッターエリアに印刷することもできます。このような計算は統計関数を使用して実行します (統計関数参照)。

## 出力コントロールラインを使用する

ヘッダー、詳細、ブレーク、フッターエリアはフォームエディターのエリアマーカーを使用して制御します。各エリアのマーカーを縦に移動して高さを増減できます。各エリア内に配置されたオブジェクトは、対応する場所に表示や印刷されます。




**マーカーとエリア**

一覧フォームがアプリケーションモードやデザインモードで表示されたとき、各エリアは以下のように動作します。

- **ヘッダーエリア**: ヘッダーエリアは画面の一番上に表示され、また印刷時には各ページの一番上に印刷されます。ヘッダーエリアはフォームの一番上からヘッダーマーカー (H) までの間です。ヘッダーマーカーを上下に移動して高さを変更できます。ヘッダーエリアには列のタイトル、フォームの説明、その他の情報、会社ロゴなどの画像を配置します。
  サブフォームとして表示される一覧フォーム、あるいはDISPLAY SELECTION and MODIFY SELECTIONコマンドを使用して表示される一覧フォームのヘッダーエリアにアクティブオブジェクトを配置して使用することもできます。
  以下のようなアクティブオブジェクトを配置できます:
  - ボタン、3Dボタン、ハイライトボタン、ピクチャーボタン
  - コンボボックス、ポップアップメニュー/ドロップダウンリスト、ピクチャーポップアップメニュー、階層ポップアップメニュー
  - 階層リスト、リストボックス
  - ラジオボタン、3Dラジオボタン、ピクチャーラジオボタン
  - チェックボックス、3Dチェックボックス
  - サーモメーター、ルーラー、ダイアル

  サブレコード追加やくキャンセルなどの標準アクション、自動スプリッターをボタンに割り当てることができます。ヘッダーエリアに配置したアクティブオブジェクトでは以下のイベントを使用できます: On Load, On Clicked, On Header, On Printing Footer, On Double Clicked, On Drop, On Drag Over, On Unload。フォームメソッドはエリアのオブジェクトメソッドが呼び出された後、On Headerイベントで呼び出されます。

- **詳細エリア**: 詳細エリアはレポート中の各レコード毎に画面に表示されたり印刷されたりします。詳細エリアはヘッダーマーカー (H) と詳細マーカー (D) の間です。詳細マーカーを縦に移動して高さを変更できます。詳細エリアに配置したオブジェクトは各レコード毎に表示または印刷されます。主にフィールドや変数を配置して各レコードの情報を表示/印刷しますが、四角など他のオブジェクトを配置することもできます。

- **ブレークエリア**: ブレークエリアはレコードリストの下に一回表示、あるいは印刷されます。ブレークエリアは詳細マーカー (D) とブレークマーカー (B0) の間です。レポートにブレークエリアを追加することもできます。ブレークマーカーを縦に移動して高さを変更できます。レコードデータではない情報 (説明や日付、時刻など) や線、その他のグラフィック要素を表示するためにブレークエリアを使用できます。印刷レポートでは小計などのサマリ計算を行って印刷するために使用できます。

- **フッターエリア**: フッターエリアはレコードリストの下の画面に表示されます。またレポートの各ページの一番下に印刷されます。フッターエリアはブレークマーカー (B0) とフッターマーカー (F) の間です。フッターマーカーを縦に移動して高さを変更できます。
  フッターエリアを使用して画像、ページ番号、日付、また各ページの下部に配置したいテキストを何でも印刷できます。画面用の一覧フォームではフッターエリアに検索や並び替え、印刷などを行うためのボタンを配置できます。すべてのアクティブオブジェクトを配置できます。

フォームが一覧フォームとして使用されるときは、画面用であれ印刷用であれ、マーカー設定が考慮されて各エリアが指定された場所に表示/印刷されます。また出力マーカーはフォームがサブフォームとして使用される場合も有効です。

フォームが詳細フォームとして使用される場合、マーカー設定は無視されます。

これらのエリアに配置されたオブジェクトに割り当てられたメソッドは、適切なイベントが有効にされていれば、エリアが表示/印刷されるときに実行されます。例えばヘッダーエリアに配置されたオブジェクトのメソッドはOn Headerイベントで実行されます。

レポート用に追加のブレークエリアやヘッダーエリアを作成できます。これらの追加されたエリアを使用して小計を印刷したりできます。追加のエリアについては追加マーカーの作成を参照してください。

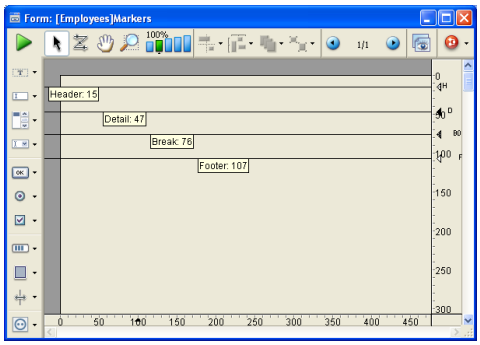
**マーカーの移動**

ヘッダー、詳細、ブレーク、フッターエリアの高さはマーカーを移動して変更できます。

マーカーはフォームを横切る線として表示されます。各マーカーはマーカーラベルおよびルーラー上に表示されるタイトルで識別できます。ルーラー上には三角とタイトルが表示され、これらをクリックしてドラッグすると、マーカーが移動します。デフォルトではマーカーラベルが表示されています。マーカーやマーカーラベルは非表示にすることもできます (フォームエディター参照)。

各マーカーの名前と位置を示すラベルはフォームの起点から計算されます。マーカーを移動するとラベル上には新しい位置がリアルタイムに表示されます。

以下の図はマーカーとそのラベルです。:




マーカーを移動するにはルーラー上の三角またはマーカーラベルを縦にドラッグします。

- マーカーをドラッグ中にShiftキーを押したままにすると、そのマーカーより下のマーカーも一緒に移動できます。例えばShiftキーを押しながらヘッダーマーカーを移動すると、すべてのマーカーが移動されます。ヘッダーマーカー以外のマーカーを一緒に移動する場合、Shiftキーを押しながら詳細マーカーをドラッグします。
- マーカーに移動とともに、マーカーより下にあるオブジェクトを移動したり、マーカーより上のオブジェクトを縦に拡大したりするには、マーカーに移動を開始する前に対象のオブジェクトを選択します。

マーカーを他のマーカーを超えて移動することはできません。例えばフッターマーカーをブレークマーカーより上に移動することはできません。

マーカーを他のマーカーに重ねることはできます。この場合、そのエリアの高さは0になり、レポートから取り除かれます。例えばブレークエリアに印刷するものがない場合、ブレークマーカーを詳細マーカーに重ねてその高さを0にできます。

詳細エリアを印刷する必要がない場合、詳細マーカーをヘッダーマーカーに重ねます。ヘッダーが必要ない場合、ヘッダーマーカーを一番上 (0の位置) までドラッグします。

**警告:** 詳細エリアに配置するフィールドや変数はヘッダーやフッターエリアにはみ出してはいけません。はみ出すとフォームが実行されたときに表示されなくなります。

**追加マーカーの作成**

この節で使用する例題ではブレークおよびブレークヘッダーを使用しています。これらのエリアを追加するには、マーカー追加を行います。

フォームエディターを新規に作成したときにはH, D, B0, そして Fマーカーがあります。

B0は"ブレークレベル0"を表します。レベル0はレポート対象の全レコードが印刷された後に実行されます。追加のブレークマーカーは番号で指定されます。B1マーカーは"ブレークレベル1"を表します。
レベル1のブレークは一番目のソートフィールドでグループ化されたレコードが印刷された後に発生します。

| ラベル | 説明 | 実行タイミング |
|---|---|---|
| B1 | ブレークレベル1 | 一番目のソート条件フィールド印刷後 |
| B2 | ブレークレベル2 | 二番目のソート条件フィールド印刷後 |
| B3 | ブレークレベル3 | 三番目のソート条件フィールド印刷後 |

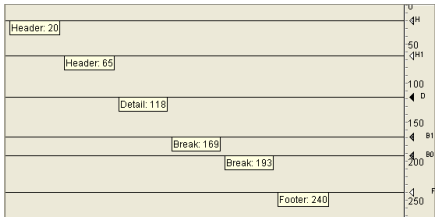
Hは"ヘッダー"を意味し、各ページの先頭に印刷されます。追加のヘッダーマーカーはブレークに関連付けられます。H1は"ヘッダーレベル1"であり、一番目のソートフィールドでグループ化されたレコードが印刷される前に発生します。

| ラベル | 説明 | 実行タイミング |
|---|---|---|
| H1 | ヘッダーレベル1 | 一番目のソート条件フィールド印刷前 |
| H2 | ヘッダーレベル2 | 二番目のソート条件フィールド印刷前 |
| H3 | ヘッダーレベル3 | 三番目のソート条件フィールド印刷前 |

マーカーの追加は**Alt**キー (Windows) または**Option**キー (Mac OS) を押しながら、マーカーをクリックすることで行います。ブレークマーカーをクリックすると追加のブレークエリアが作成されます。ヘッダーマーカーをクリックするとブレークヘッダーが作成されます。新しいマーカーは既存のマーカーの後ろに作成されます。新しいマーカーを表示するには、既存のマーカーを移動しなければなりません。

ブレーク処理を開始するためにSubtotalコマンドを使用する場合、ソート順に従って作成されるすべてのブレークレベル数から1マイナスした数のブレークエリアを作成すべきです。ブレークエリアに何も印刷する必要がない場合、マーカーを移動してその高さを0にします。ブレークエリアの数よりも多いフィールド数でソートすると、印刷時に最後のブレークエリアが繰り返されます。

下図はマーカーを追加した状態を示しています:




**マーカーの削除**

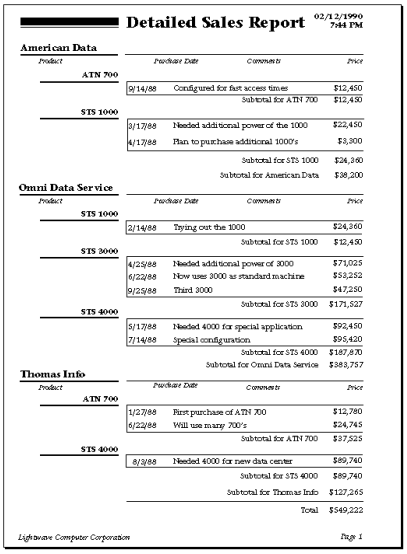
追加したブレークやブレークヘッダーを削除するには、**Ctrl**キー (Windows) または **Command**キー (Mac OS) を押しながらブレーク、ブレークヘッダー、あるいはそのラベルをクリックします。

4Dはマーカーを削除し、必要であればマーカーの番号を振り直します。最初からあるマーカー (H, D, B0, F) は削除できません。

## レポートの例題

ここでは例題を示し、フォームエディターでどのようにそのフォームを設計するのか、そして印刷をコントロールするためのメソッドを紹介します。

印刷されるレポートは以下のようになります:



Detailed Sales Report 02/12/1990 7:44 PM

| Product | Purchase Date | Comments | Price |
|---|---|---|---|
| **American Data** | | | |
| 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** | | | |
| 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** | | | |
| 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 |

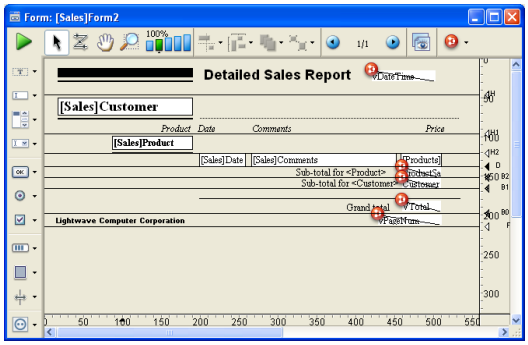
Lightwave Computer Corporation Page 1

この例題ではページヘッダーに日付、時刻、そしてレポートのタイトルが印刷されます。ブレークヘッダーには顧客名と、最初のブレークで提示される情報のための列タイトルが印刷されます。詳細エリアにはレコードから取得した情報が直接印刷されます。レベル2のブレークには各顧客ごとかつ製品ごとの小計が印刷されます。レベル1のブレークには各顧客ごとの小計が、レベル0のブレークにはレポートの総計が印刷されます。フッターにはページ番号が印刷されます。

レポートはブレークレベル数+1でソートします。このレポートでは顧客、製品、そして日付でソートします。

### レポートフォーム

下図は先に示したレポートを作成するためのフォームをエディターで開いたものです:




フォーム中の各マーカーは、対応するエリアの底を示しています。フォーム中に配置されたオブジェクトはそれが配置されたエリアに基づいて、レポート中の適切な場所に印刷されます。ヘッダーには各ページの先頭に印刷する要素を配置します。詳細エリアには各レコード毎に繰り返し印刷される要素を配置します。

各マーカーの意味は以下の通りです:

| ラベル | 説明 | 効果 |
|---|---|---|
| H | ヘッダーエリア | 各ページの先頭に一回印刷されます |
| H1 | レベル1ヘッダーエリア | レベル1ブレークの前に一回印刷されます |
| H2 | レベル2ヘッダーエリア | レベル2ブレークの前に一回印刷されます |
| D | 詳細エリア | セレクション中の各レコード毎に印刷されます |
| B2 | レベル2ブレークエリア | レベル2ブレークの後に一回印刷されます (二番目のソートフィールドの値が変わったとき) |
| B1 | レベル1ブレークエリア | レベル1ブレークの後に一回印刷されます (一番目のソートフィールドの値が変わったとき) |
| B0 | レベル0ブレークエリア | レポートの最後に一回印刷されます |
| F | フッターエリア | 各ページの最後に一回印刷されます |

### レポートのオブジェクトメソッド

ヘッダー、ブレーク、そしてフッターエリアに配置された入力不可オブジェクトはオブジェクトメソッドを使用して制御できます。

**注:** 以下のコードはオブジェクトメソッドまたはフォームメソッドでのみ利用できます。プロジェクトメソッドでは使用できません。

日付はヘッダーエリアに配置されたvDate変数のオブジェクトメソッドでシステム日付を取得します:

```
vDate:=Current date
```

時刻はヘッダーエリアに配置されたvTime変数のオブジェクトメソッドでシステム時刻を取得します:

```
vTime:=Current time
```

レベル2ブレークエリアの売上小計は、vSalesProdの以下のメソッドを使用して計算されます:

```
vSalesProd:=Subtotal(Sales)
```

レベル1ブレークエリアの売上小計は、vSalesCustの以下のメソッドを使用して計算されます:

```
vSalesCust:=Subtotal(Sales)
```

レベル0ブレークエリアの売上総計は、vSalesTotalの以下のメソッドを使用して計算されます:

```
vSalesTotal:=Subtotal(Sales)
```

上の3つのオブジェクトメソッドは同じ計算式を使用していますが、それぞれ結果は異なります。それぞれのオブジェクトは異なるブレークエリアに配置されているため、各コードは異なるタイミングで異なるレコードグループに対して行われます。ブレークレベルについてはレポート印刷用のフォームの"ブレークレベルを使用する"を参照してください。

```
vPage:="Page "+String(FORM Get current page)
```

FORM Get current page関数はページ番号を返します。

## 可変サイズエリアの印刷

レポート中でサブフォームやピクチャー、テキストを印刷できます。これらのオブジェクトは固定長あるいは可変長のフレームで印刷できます。固定長フレームのオブジェクトは、それがフォーム上に作成された通りの大きさに印刷されます。可変長フレームのオブジェクトは、オブジェクトの内容がすべて印刷されるよう、印刷時に拡大されます。

フォーム上で可変長オブジェクトを横に並べ置くことはできません。ピクチャーやテキスト可変長オブジェクトの横に固定長オブジェクトを配置することができますが、この場合可変長オブジェクトの高さはその横に配置されたオブジェクトよりも最低一行高く設定されていて、またそれらすべてのオブジェクトの上端が揃えられていなければなりません。この条件を満たさない場合、印刷時に可変長オブジェクトフレームの高さ分、他のフィールドの値が繰り返し印刷されます。

可変長サブフォームの両側にオブジェクトを配置することはできません。
可変長オブジェクトは詳細エリアにのみ配置できます。

### サブフォームエリアの印刷

サブフォームにレコードを印刷する場合、通常は一覧フォームを使用します。

サブフォームの印刷オプションを設定するには、プロパティリストでサブフォームプロパティの印刷グループを表示します。

レコードがサブフォームエリアに収まりきらない可能性があるため、サブフォームの印刷には以下3つの選択肢があります:

- **可変**:4Dは全レコードが印刷されるようサブフォームエリアを拡大/縮小します。
- **固定 (切り捨て)**: 4Dはサブフォームエリアに収まるだけのレコードを印刷します。フォームは一回だけ印刷され、印刷されなかったレコードは無視されます。
- **固定 (全レコード)**: フレームは指定されたサイズで印刷されますが、4Dは全レコードが印刷されるまでそのフォームを複数回印刷します。

**注:** 可変長サブフォームの両側にはオブジェクトを配置できません。配置するとサブフォームの各行毎にそのオブジェクトが繰り返し印刷されます。

サブフォームオプションに関する詳細はリストサブフォームを参照してください。

### ピクチャータイプオブジェクトの印刷

表示フォーマットが対応していれば、ピクチャーを固定長にも可変長にも印刷できます。以下のフォーマットが可変長フレームに対応しています:

- トランケート (中央合わせあり)
- 背景
- トランケート (中央合わせなし)

ピクチャーフォーマットについては表示フォーマットを参照してください。

ピクチャー印刷オプションを設定するには、ピクチャーフィールドや変数のプロパティリストで印刷グループを表示します。そして**プリント時可変**チェックボックスを必要に応じて設定します:

- このオプションを選択すると、ピクチャーはその高さの通りに印刷されます。必要に応じてピクチャーフレームが拡大されます。
- このオプションを選択しないと、ピクチャーはフォームに設定された固定の高さで印刷されます。

### テキストタイプオブジェクトの印刷

テキストフィールドを固定長あるいは可変長で印刷できます。テキストフィールドの設定はピクチャーフィールドの設定と同じです。

フォームエディター中でテキストフィールドのプロパティリストで印刷グループを表示し、必要に応じて**プリント時可変**オプションを選択します:

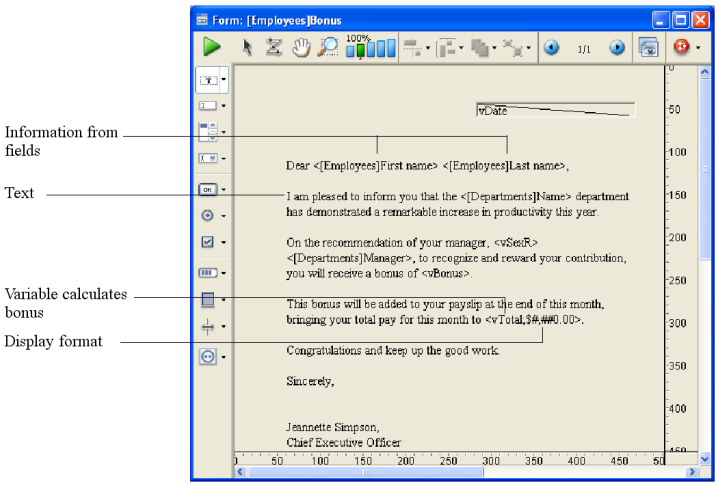
- このオプションを選択すると、テキストはその高さの通りに印刷されます。すべてのテキストが印刷されるよう、必要に応じてフレームが拡大されます。
- このオプションを選択しないと、テキストはフォームに設定された固定の高さで印刷されます。

## 差し込み印刷ドキュメントを作成する

出力フォーム中でスタティックテキストエリアにフィールドの値、フィールドやテーブルのタイトル、変数の値を埋め込んで印刷できます。名前や住所、計算値だけが異なるテンプレートフォームを作成できます。差し込み印刷は複数の宛先に同じ内容の郵便物を送信するような場合に便利です。

差し込み印刷を行う場合、フォームウィザードで**印刷用詳細フォーム**フォームタイプを選択します。フォームエディターではテキストエリアを作成し、そこにスタティックテキストの他、フィールドや変数への参照を埋め込みます。印刷時にはフィールドや変数への参照が実際の値に置き換えられます。

以下の図はテキストエリアにフィールドと変数が埋め込まれた状態を示しています。




差し込み印刷用の出力フォームを作成するには、フォームタイプを"印刷用詳細フォーム"に設定し、詳細エリアにスタティックテキストエリアを配置します。

スタティックテイストエリアには印刷したい内容を入力します。フィールドや変数の名前は (<) や (>) 記号の間に記述します。こうすることにより、印刷時に実際の値に置き換えられます。

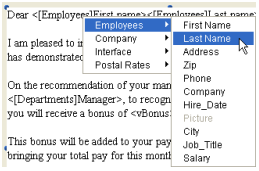
データベース中のどのフィールドも参照できます。マスターテーブルのフィールドを参照する場合、テーブル名は省略でき、<Field> のように記述できます。他のテーブルのフィールドを参照する場合はテーブル名を指定しなければならず、<[TableName]Field> のように記述します。

テーブルのラベルを印刷するには <?[TableName]> や <?[N]> (Nはテーブル番号) のように記述します。フィールドラベルを印刷するには <?[TableName]FieldName> や <?[X]Y> (Xはテーブル番号でYはフィールド番号) のように記述します。詳細はスタティックテキスト中で参照を使用するを参照してください。

変数にはオブジェクトやフォームメソッドを使用して値が割り当てられていなければなりません。

埋め込むフィールドや変数の印刷フォーマット (GET LIST ITEM PARAMETER参照) を指定する場合、参照の後ろにセミコロンそして印刷フォーマットを記述します。例えば先の例ではvTotal変数を <vTotal;$###,##0.00> で埋め込んでいて、計算された値は$通貨フォーマットで印刷されます。

テキストエリアにフィールドを挿入するショートカットが用意されています。ショートカットを使用すればドロップダウンメニューからフィールド名を選択できます。これを行うにはフィールド参照を埋め込む場所にカーソルを置き、**Alt**キー (Windows) または**Option**キー (Mac OS) を押しながらマウスをクリックします。
4Dはマスターテーブルのフィールドをポップアップメニューに表示します。
他のテーブルのフィールドを挿入するには、**Shift+Alt** (Windows) または**Shift-Command** (Mac OS) を押しながらマウスをクリックします。4Dはテーブルとフィールドの階層メニューを表示します。




フィールドを選択すると4Dは <Field> または <[Table]Field> をテキストエリアのカーソル位置に挿入します。

レポートが印刷される際、フィールドや変数の参照部分が実際の値に置き換えられます:

## ラベルの作成

ラベルやカスタムレポートフォームを使用してラベルを生成できます。レポートフォームを使用する場合、よりカスタマイズを行うことが可能です。特にフォームエディターを使用してラベル中に変数を挿入することができるようになります。

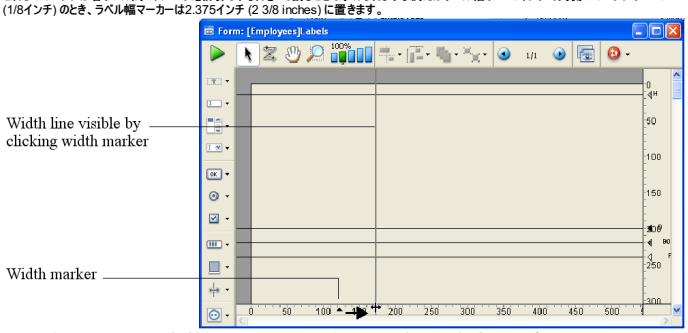
フォームエディターでラベルレポートフォームを作成したら、以下の2つの方法で使用できます:

- PRINT LABELコマンドを使用する
  PRINT LABEL([myTable]) 文を実行するとテーブルのカレントセレクションがカレントの一覧フォームで印刷されます。この場合4Dはラベルのフォーマット (高さと幅) を設定するためにマーカーを使用します。
- ラベルエディターを使用する
  この方法を使用すると、フォームエディターの詳細機能とラベルエディターのレイアウトパラメーター両方の利点を利用できます。
  ラベルエディターはフォーム上のオブジェクトの絶対座標と設定されたマージンのみを考慮します。マーカーは無視されます。

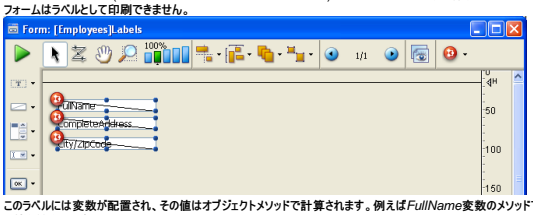
### PRINT LABELコマンドを使用する

ラベルレポートフォームを作成しそれをPRINT LABELコマンドで使用するには:

1. ラベル幅をフォームエディターの横ルーラー上の幅マーカーで設定します。
   幅マーカーにより、PRINT LABELコマンドがページに印刷するラベルの横の数が決定されます。幅マーカーの位置はラベルの幅に対応しなければなりません。ラベルの左右マージンを考慮に入れてください。
   これらのマージンは各ラベル間のスペースを計算し、それを2で割ることで求められます。例えばラベル幅が2.25インチで両側のマージンが.125 (1/8インチ) のとき、ラベル幅マーカーは2.375インチ (2 3/8 inches) に置きます。



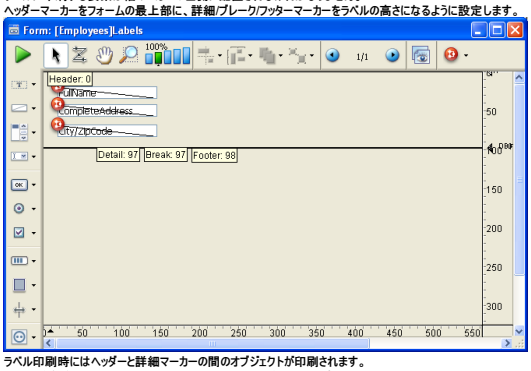

   **注:** ラベル幅マーカーの配置をしやすくするために、ルーラーの単位を戦地やインチに変更できます (デフォルトはポイント)。これを行うには**フォーム**メニューから**ルーラー定義**を選択します。
2. ラベルを構成するオブジェクトを配置します。
   ラベルにはフィールドや (必要に応じてメソッドを記述することができる) アクティブオブジェクト、画像、スタティックテキストを配置できます。サブフォームはラベルとして印刷できません。




   このラベルには変数が配置され、その値はオブジェクトメソッドで計算されます。例えば*FullName*変数のメソッドではレコードの姓と名フィールドの値を結合します:

```
FullName:=[Customers]FirstName+" "+[Customers]LastName
```

3. 幅マーカーに収まるようにオブジェクトを配置します。
   ラベルに印刷する要素は幅マーカーの左側に配置されなければなりません。
4. ヘッダーマーカーをフォームの最上部に、詳細/ブレーク/フッターマーカーをラベルの高さになるように設定します。

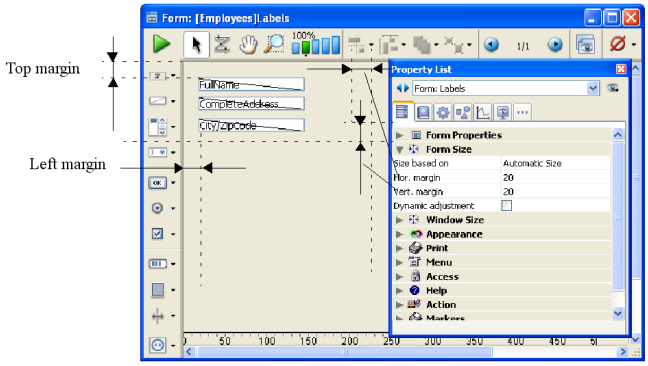



   ラベル印刷時にはヘッダーと詳細マーカーの間のオブジェクトが印刷されます。
5. フォームを保存し、PRINT LABELコマンドでセレクションを印刷します。

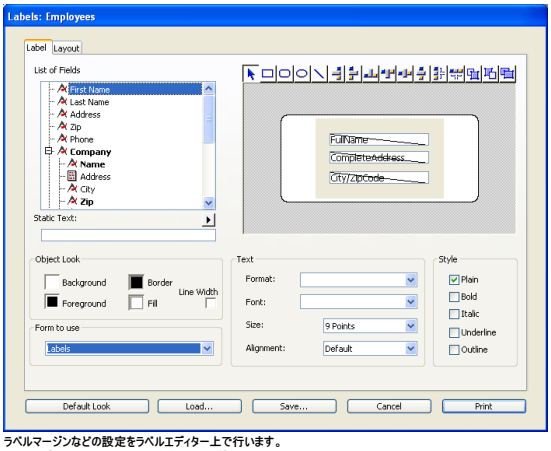
### ラベルエディターを使用する

ラベルレポートフォームをラベルエディター内で使用する手順は以下の通りです:

1. フォームエディター上でラベルを構成するオブジェクトを挿入します。
2. 以下の原則に基づき各要素を配置します:
   - 各ラベルの左上マージンは、フォーム座標(0,0)からの絶対座標で決定されます。
   - 各ラベルの右下マージンはフォームのプロパティリストの**横マージン**と**縦マージン**により決定されます。




3. テーブルリストウィンドウで対象のテーブルを選択します。
   テーブルリストウィンドウが表示されていない場合、**レコード**メニューから**テーブルリスト**コマンドを選択します。
4. **ツール**メニューから**ラベル**を選択します。
   ラベルエディターが表示されます。
5. "利用するフォーム"ドロップダウンリストからラベルフォームとして使用するフォームを選択します。
   これにより4Dはラベル印刷にこのフォームを使用します。




6. ラベルマージンなどの設定をラベルエディター上で行います。
   ラベルエディターについてはラベルを参照してください。
7. **プリント**ボタンをクリックします。
   4Dは指定されたラベルデザインを使用してカレントセレクションを印刷します。

# ユーザーフォーム


ユーザーフォームの概要 (アプリケーションモード)

ユーザーフォームエディター

オブジェクトの表示と編集

データ入力順

オブジェクトライブラリの利用

## ユーザーフォームの概要 (アプリケーションモード)

開発者はデータベース中の特定のフォームを、ユーザーが更新できるよう許可できます。これによりユーザーはカスタムレポートを作成したり、ロゴを追 加したりなど、必要に応じてフォームを更新できます。更新が保存されると、そのユーザーマシン上ではユーザーフォームが元のフォームの代わりに使用されます。

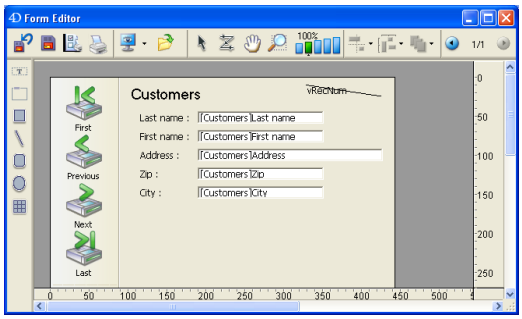
変更可能なのは主にグラフィックに関する部分です。例えばユーザーがオブジェクトメソッド、変数やフィールドを追加することはできません。しかし入 力順を変更したり、オブジェクトライブラリからアクティブオブジェクトを追加したりすることができます。デザインモードのフォームエディタを使用して、開 発者はユーザーに変更を許可する範囲を指定できます。

開発者がデザインモードでこの仕組みを設定するメカニズムは4Dランゲージリファレンスマニュアルのユーザフォームで説明されています。
この章ではアプリケーションモードでのユーザーフォームエディターの使い方について説明します。

## ユーザーフォームエディター

ユーザーフォームエディターはEDIT FORMコマンドが実行され、対象フォームの**ユーザーによる編集可**プロパティが選択されている場合に表示されます (フォームプロパティ参照)。

エディターウィンドウは以下のように表示されます:




このエディターは4D標準のフォームエディターに似ており、ツールバーとオブジェクトバーを持ちます。またこのエディター特有の機能もあります。

エディターが表示されると、メニューバー中で**編集**メニューだけが利用可能となります。簡易化されたプロパティリストとコンテキストメニューが利用できます。

オブジェクトの選択と移動は (許可されていれば) 標準のフォームエディターと同じ方法で行えます。

4D標準のフォームエディターについてはフォームエディターを参照してください。

### ツールバー

ツールバーには以下の項目が含まれます:

- **戻す**: フォームが最後に保存された状態に戻します。このボタンをクリックすると警告ダイアログボックスが表示され、処理を続行するかキャンセルするか選択できます。
- **保存**: ユーザーフォームに対して行った変更を保存します。
- **印刷設定**と**印刷**: これら2つのボタンを使用してユーザーフォームの印刷設定を行ったり印刷を開始したりできます。
- **表示**: ポップアップメニューを使用してエディター上に表示する項目を選択できます:
  - **ページ0**: 0ページ上のオブジェクトをフォームのカレントページに表示するか非表示にするか選択します。このオプションを使用してカレントページ上のオブジェクトと0ページ上のオブジェクトを簡単に見分けることができます。複数ページのフォームでは0ページ上のオブジェクトがすべてのページに表示されます。
  - **用紙**: 印刷用用紙サイズの境界線 (灰色の線) の表示/非表示を切り替えます。
  - **ルーラー**: フォームエディターウィンドウのルーラーの表示/非表示を切り替えます。
  - **マーカー**: 出力マーカー線とそのラベルの表示/非表示を切り替えます。マーカーは一覧フォームで使用され、ヘッダー、詳細、フッターエリアを指定するものです。
  - **境界**: フォーム境界の表示/非表示を切り替えます。このオプションが選択されていると、フォームは画面に表示されるサイズの通りにエディター中に表示されます。
- **ライブラリ**: フォームで利用可能なオブジェクトライブラリをロードするために使用します (カスタムオブジェクトライブラリの作成と使用参照)。
- **選択ツール**: フォームオブジェクトの選択、移動、サイズ変更に使用します。
- **入力順**: "入力順"設定モードに移動します。このモードでは現在の入力順を表示し、それを変更できます (データの入力順を変更する参照)。
- **移動**: フォーム移動モードに移行します。このモードではウィンドウ中をドラッグ＆ドロップで素早く移動できるようになります。カーソルは手の形になります。このナビゲーション方法はフォームを拡大表示しているとき特に便利です。
- **拡大**: フォームの拡大表示モードを切り替えます (デフォルトで100%)。虫眼鏡ボタンをクリックして拡大モードにするか、直接拡大率バーをクリックします。
  このボタンをクリックするとカーソルは虫眼鏡の形になります。1レベル拡大するためにはフォーム中をクリックします。拡大レベルを下げるにはShiftキーを押しながら (カーソルが に変わります) フォーム中をクリックします。
- **揃え**: このボタンはフォーム中でオブジェクトを揃えるメニューコマンドにリンクされています。このメニューを有効にするには2つ以上のオブジェクトが選択されていなければなりません。
- **整列**: このボタンはフォーム中でオブジェクトを整列するメニューコマンドにリンクされています。このメニューを有効にするには2つ以上のオブジェクトが選択されていなければなりません。
- **レベル**: このボタンはフォーム中でオブジェクトのレベルを変更するメニューコマンドにリンクされています。
- **ページ**: このエリアを使用してフォームエディター中で表示ページを切り替えられます。ページを切り替えるには左右の三角ボタンをクリックするか、中央のエリアで表示するページを直接選択します。ユーザーフォームエディターでページを追加することはできません。

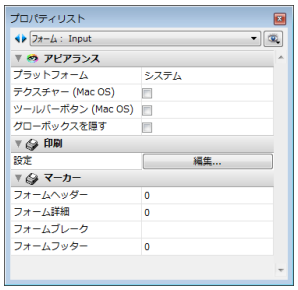
### オブジェクトバー

ユーザーフォームエディターのオブジェクトバーを使用して、様々なグラフィックオブジェクト (**テキスト**、**グループボックス**、**四角**、**線**、**角の丸い四角**、**楕円**、**格子**) をフォームに追加できます。

フォーム中にオブジェクトを作成するには、対応するボタンを選択し、フォーム中にオブジェクトを描画します。

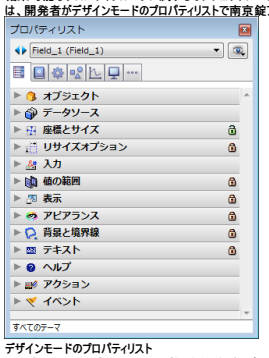
## オブジェクトの表示と編集

簡易化されたプロパティリストを使用して、ユーザーはフォーム中のオブジェクトのプロパティを確認したり編集したりできます。プロパティリストを表示するにはオブジェクトをダブルクリックするか、エディタのコンテキストメニューから**プロパティリスト**を選択します。

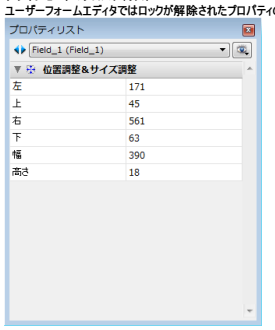



ユーザーが編集可能なプロパティのみがプロパティリストに表示されます。

- オブジェクトバーを使用してユーザーが追加したグラフィックオブジェクトでは、そのオブジェクトタイプにリンクされたすべてのプロパティを利用できます。
- 元のフォームに開発者が配置したオブジェクトでは、ユーザーがそのプロパティを編集できるには2つの条件があります:
  - そのプロパティが編集可能なプロパティグループに属していること。例えばフォームイベント管理やドラッグ＆ドロップに関連するプロパティはこのグループに属しておらず、ユーザーは編集できません。デザインモードのフォームエディターで、ユーザー編集可能なプロパティには南京錠アイコンがついています (フォームプロパティの"ユーザー編集可"参照)。
  - 編集可能なプロパティグループに属するオブジェクトについて、開発者が明示的にユーザーに編集を許可していること。編集を許可するには、開発者がデザインモードのプロパティリストで南京錠アイコンをクリックし、鍵が解除された状態にしなければなりません:




デザインモードのプロパティリスト
ユーザーフォームエディタではロックが解除されたプロパティのみアクセスできます:




上記の場合"位置調整＆サイズ調整"プロパティがロック解除されているため、ユーザーはオブジェクトを移動したり大きさを変更したりできます。

### 編集メニューコマンド

ユーザーフォームエディターウィンドウがデザインモードで表示されているとき、**編集**メニューのみが利用可能です。アプリケーションモードでカスタマイズインターフェースを使用している場合、メニューに含まれるコマンドが標準アクションで管理されている場合、**編集**メニューにアクセスできます。

すべての編集コマンド (**カット**, **コピー**, **ペースト**, **クリア**, **すべてを選択**) は、ユーザーが追加したオブジェクトに対しては制限なく動作します。

他方デザイナーが配置したオブジェクトをユーザーが削除することはできません。**カット**や**クリア**コマンド、または**Del**や**Backspace**キーを使用すると、その動作が許可されない旨警告するダイアログが表示されます。

### コンテキストメニュー

いくつかのコマンドにはユーザーフォームエディターのコンテキストメニューからアクセスできます。このメニューの内容を変更することはできません。

#### オブジェクト以外の場所をクリックした場合

オブジェクトが存在しない場所でコンテキストメニューを表示させると、フォームエディターの標準表示や編集コマンドにアクセスできます:

- **マグネティックグリッド**: マグネティックグリッドを使用すると、フォーム上でオブジェクトを揃えたり整列したりすることを簡単に行えます。
  マグネティックグリッドは以下のように動作します: フォーム中でオブジェクトをドラッグ＆ドロップすると、他のオブジェクトの座標に基づき、4Dがガイド線を表示します:
  - 2つのオブジェクトの端または中央が重なったところで横線が引かれます。
  - 2つのオブジェクトの端が重なったところで縦線が引かれます。

  この時点で4Dは揃えあるいは整列が行われていることを示す赤線を表示し、オブジェクトをその位置に吸着させます:
  - 上端揃え
  - 中央揃え

  オブジェクトの整列の場合、4Dはインターフェース標準に基づく間隔を提案します。マグネティック揃えと同様、赤線を使用して適切な場所が示されます。
  **注: Ctrl** (Windows) または **Control** (Mac OS) を押しながらドラッグを行うと、一時的にマグネティックグリッドを有効/無効にできます。

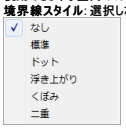
- **表示**: ここにはエディターツールバーの**表示**メニューと同じものが表示されます (ユーザーフォームエディター参照)。

#### オブジェクト上をクリックした場合

ユーザーがオブジェクト上でコンテキストメニューを表示した場合、カラーや整列などのグラフィックコマンドを利用できます。

元のフォームに属するオブジェクトについてはこれらのコマンドへのアクセスが制限されることがあります。これはクリックされたオブジェクトのタイプおよび開発者がフォームエディターで設定した内容に基づきます。

- **カラー**: オブジェクトの描画色と背景色を変更するためのパレットを2つ表示します。
- **整列**: オブジェクトの揃えや整列を行うためのコマンドを提供します。揃えコマンドは2つ以上の移動可能なオブジェクトが選択されている場合に使用できます。整列コマンドは3つ以上の移動可能なオブジェクトが選択されている場合に使用できます。
- **境界線スタイル**: 選択したオブジェクトの境界線スタイルを変更するために使用します。利用可能なスタイル例は以下の通りです:

  *なし / 標準 / ドット / 浮き上がり / くぼみ / 二重*
- **最適サイズ**: このオプションを使用すると、テキストの場合現在のグラフィック特性 (フォントサイズやスタイル等) に基づいて、ピクチャーの場合元のサイズでピクチャーを表示するように、オブジェクトのサイズを最適に変更できます。同じ結果はオブジェクトの右下リサイズハンドルを**Ctrl** (Windows) または **Command** (Mac OS) + クリックすることでも得られます。
- **レベル**: オブジェクトのフォーム中でのZ座標 (重なり順) を設定するために使用します。
- **同種のオブジェクトを選択**: 選択されたオブジェクトと同じタイプのオブジェクトをすべて選択するために使用します。

## データ入力順

データ入力順とはフィールド、サブフォーム、そしてその他のフォーカス可オブジェクトが、タブキーによって選択される順番のことです。

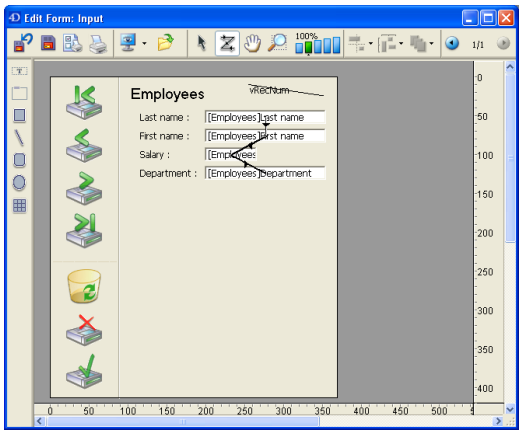
**注:** フォームでのデフォルト入力順については[データの入力順を変更する]を参照してください。

オブジェクトを追加したときなど、入力順をより論理的な順番に変更したくなることがあります。

### 入力順の表示と変更

現在のデータ入力順を表示するにはツールバーの**入力順**ボタンをクリックします:

現在のデータ入力順は以下のように表示されます:




- データ入力順を変更するにはフォーム上のオブジェクト上にマウスポインターを合わせ、マウスボタンを押しながら次の入力順にするオブジェクトまでドラッグします。
- データ入力順の変更が終了したら**オブジェクト選択**ボタンをクリックして入力順表示モードを解除します。

### データ入力順の最初のオブジェクトを設定する

すべての入力可オブジェクトはデータ入力順に組み込まれます。

入力順の最初のオブジェクトを設定するには以下の手順を行います:

1. **選択ツール**ボタン をクリックして入力順モードを解除します。
2. 入力順の先頭にするオブジェクトを選択し、レベルを背面にします。
   コンテキストメニューから**レベル > 背面へ移動**コマンドを選択するか、エディターツールバーの対応するボタンを使用します。オブジェクトは他のオブジェクトよりも背面におかれます。これは一時的なものです。
3. **入力順**ボタン をクリックします。
   選択したオブジェクトが入力順先頭になります。今まで先頭だったオブジェクトは2番目になります。ドラッグを使用してその他の入力順を設定できます。
4. 入力順の設定が終了したら**選択ツール**ボタンをクリックして入力順モードを解除します。必要に応じて背面にしたオブジェクトを元に戻します。

### データ入力グループを使用する

データ入力順を変更中、フォーム上でオブジェクトグループを選択し、そのグループ内で標準のデータ入力順を適用することができます。これによりグループや列に分離されているフィールドに対し、データ入力順を簡単に設定できます。

グループでデータ入力を設定するには:

1. ツールバーの**入力順**ボタンをクリックします 。
2. データ入力グループを構成したいオブジェクトの周りをマウスでドラッグしてマーキーを描画します。
   マウスボタンを離すと、マーキー内およびマーキーに重なったオブジェクトに標準のデータ入力順が適用されます。必要に応じて他のオブジェクトのデータ入力順も調整されます。

### 標準のデータ入力順に戻す

いつでも標準の入力順に戻すことができます。

1. **入力順**ボタンをクリックします 。
2. フォーム上のすべてのオブジェクトを囲むようにマウスでマーキーを描画します。
   マウスボタンを離すと4Dはすべてのオブジェクトに対し標準のデータ入力順を適用します。

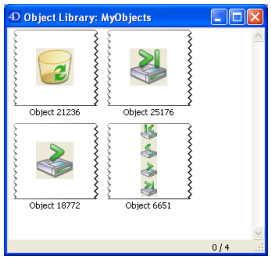
## オブジェクトライブラリの利用

変更を行う各ユーザーフォームごとに、オブジェクトライブラリを使用することができます。
オブジェクトライブラリは4Dのデザインモードで作成する外部ファイルです。このファイルにはフォームで使用することのできるオブジェクト (ボタン、テキスト、階層リスト等) を格納することができます。ライブラリからドラッグ＆ドロップやコピー＆ペーストでオブジェクトをフォームに配置できます。オブジェクトライブラリについてはカスタムオブジェクトライブラリの作成と使用で説明しています。

オブジェクトはすべてのプロパティと標準アクションとともに格納されます。ユーザーフォームエディターではオブジェクトメソッドが記述されたオブジェクトを使用することができません。ユーザーフォームエディターからオブジェクトライブラリを呼び出すと、オブジェクトメソッドを持たないオブジェクトだけが表示されます。

ユーザーフォームエディターを表示するEDIT FORMコマンドの最後の引数を使用して、開発者はオブジェクトライブラリファイルを指定できます。この引数が有効であればライブラリが表示され、ユーザーフォームエディターで利用できます。ライブラリを開くには、ユーザーはエディターツールバーの対応するアイコン をクリックします。

ライブラリウィンドウが表示されます:




**注:** オブジェクトライブラリには最大32,000項目までを格納できます。

ユーザーはドラッグ＆ドロップやコピー＆ペーストを使用してフォームにオブジェクトを追加できます。ライブラリから挿入されたオブジェクトのすべてのプロパティは自動でユーザーからのアクセスが可能となります (座標、アピアランス)。

各オブジェクトライブラリはコンテキストメニューを持ち、基本的な編集コマンドへのアクセスを提供します。ユーザーフォームエディター中では、**コピー**コマンドのみを利用できます。ユーザーはライブラリから項目を削除したり名前を変更したり、ライブラリに項目を追加することはできません。

# メソッドの編集


メソッドの概要

メソッドの管理

プロジェクトメソッドプロパティ

メソッドエディター

メソッドを記述する

マクロの作成と利用

メソッド中の検索と置換

メソッドの実行

## メソッドの概要

オブジェクトのアクションを指定するために、4Dオブジェクトにメソッドを関連付けることができます。メソッドはオブジェクトが行うべきことを指示する一連の命令です。例えばメソッドを使用して以下のようなことが可能です:

- データ入力中にビジネスルールを強制する
- フィールドや変数の値を計算する
- コンボボックスや階層リスト、タブコントロールなどのインターフェースオブジェクトを管理する
- ドラッグ&ドロップアクションを管理する
- カスタムメニューコマンドにアクションを割り当てる
- 複数のプロセスを作成し管理する
- トランザクションを管理する
- カスタムレポートを管理する
- マルチユーザーデータベースアクセスを制限する

メソッドには以下のタイプがあります:

- **オブジェクトメソッド**: フォーム上の個々のオブジェクトに関連付けられます。
- **フォームメソッド**: フォームに関連付けられます。
- **トリガー**: データベースエンジンレベルで特定のイベントが発生したときに実行されます。
- **データベースメソッド**: アプリケーションの実行セッションに関連する特定のイベントが発生したときに自動で実行されます。
- **プロジェクトメソッド**: 様々な方法で呼び出されます。

メソッドタイプについては4Dランゲージリファレンスマニュアルのメソッドを参照してください。

ここでは4Dメソッドエディターを使用してメソッドを編集する方法について取り上げます。4Dのプログラム言語については4Dランゲージリファレンスマニュアルのプログラミング言語の構成要素を参照してください。

# メソッドの管理

## メソッドの作成と編集

メソッドはメソッドエディターを使用して編集します。メソッドエディターにはメソッドを作成、テスト、編集するためのツールが揃っています。
ここでは (オブジェクト、プロジェクト、トリガー、フォーム) メソッドを作成し編集する様々な方法を紹介します。
データベースメソッドはエクスプローラーからアクセスします。

**4D Serverに関する注記:** 2人以上のユーザーが同時に同じメソッドを編集しようとすると、オブジェクトのロックが発生します。あるユーザーがデザインモードで編集のためにメソッドを開くと、そのメソッドはロックされます。他のユーザーは最初のユーザーがメソッドを閉じてロックを解除するまで、そのメソッドを編集できません。ただしそのメソッドを読み込みのみモードで編集することはできます。

### オブジェクトメソッドの作成/編集

オブジェクトメソッドはフォームのオブジェクトに対して作成されます。オブジェクトメソッドを作成するオブジェクトをフォームエディター上に表示します。

フォームエディターからオブジェクトメソッドを作成/編集するには:

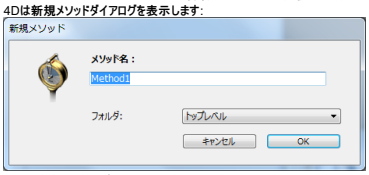
1. **Alt** キー (Windows) または **Option** キー (Mac OS) を押しながらオブジェクトをクリックする
   または
   オブジェクトを選択し、**オブジェクト**メニューから**オブジェクトメソッド**を選択する
   または
   オブジェクト上でコンテキストメニューを表示し、**オブジェクトメソッド**を選択する
   または
   オブジェクトのプロパティリストでオブジェクトメソッドの**編集...** ボタンをクリックする

メソッドエディターウィンドウが表示されます。作成した当初は空です。

### プロジェクトメソッドの作成/編集

プロジェクトメソッドを作成したり開いたりする方法は複数あります。

**ファイル**メニューからプロジェクトメソッドを作成するには:

1. **ファイル**メニューまたは4Dツールバーの**新規 > メソッド...** を選択します。
   4Dは新規メソッドダイアログを表示します:
   
   
2. メソッド名を入力します。
   メソッド名は31文字まで指定できます。使用可能な文字については4Dランゲージリファレンスマニュアルの識別子を参照してください。
3. (任意) メソッドを格納するフォルダーを選択します。
   フォルダーを選択すると、メソッドはそのフォルダーに格納されます。フォルダーはアプリケーションオブジェクトを構造化するために使用され、エクスプローラーのホームページで管理されます。デフォルトではメソッドはトップレベル (フォルダーの外) に作成されます。
4. **OK**ボタンをクリックします。
   4Dは空のメソッドエディターウィンドウを表示します。

エクスプローラーからプロジェクトメソッドを作成するには:

1. エクスプローラーのCurrent form tableページを開きます。
2. **プロジェクトメソッド**項目を選択します。
3. リストの下部にある追加ボタン をクリックします。
   新規メソッドダイアログが表示されます。
4. メソッド名を指定して**OK**をクリックします。

**ファイル**メニューからプロジェクトメソッドを開くには:

1. **ファイル**メニューから**開く > メソッド...** を選択します。
   エクスプローラーのホームページが表示されます (以下に続く)。

エクスプローラーからプロジェクトメソッドを開くには:

1. エクスプローラーのホームページページを表示します。
2. **プロジェクトメソッド**項目の階層を展開し、開くメソッド名をダブルクリックします。
   または
   メソッド名上でコンテキストメニューから**メソッド編集...** コマンドを選択します。
   そのメソッドがメソッドエディターで開かれます。

メニューバーエディターからプロジェクトメソッドを開くには:

1. メニューエディターで、カレントメニュー項目の"メソッド名"エリアをクリックします。
   このエリアには (既に割り当てられていれば) 選択されたメニュー項目に関連付けられたメソッド名が表示されています。
2. "メソッド名"ドロップダウンメニューの右にある編集ボタンをクリックします。
   メソッドがメソッドエディターウィンドウに表示されます。

メソッドエディターのウィンドウからプロジェクトメソッドを開くには:

1. メソッドエディター中で、開くプロジェクトメソッド名を選択します。
2. エディターのコンテキストメニューから**メソッド編集**を選択するか**Ctrl+K** (Windows) または **Command+K** (Mac OS)をタイプします。
   または
   プロジェクトメソッド名上で直接**Alt+ダブルクリック** (Windows) または **Option+ダブルクリック** (Mac OS) します。
   メソッドがメソッドエディターウィンドウに表示されます。4Dがメソッドを識別できない場合、エクスプローラーのホームページが表示されます。

### トリガーの作成/編集

ストラクチャーエディター内のショートカットまたはエクスプローラーからトリガーを作成できます。

ストラクチャーエディターウィンドウからトリガーを作成するには:

1. **Alt** キー (Windows) または **Option** キー (Mac OS) を押しながらストラクチャーエディターウィンドウ中のテーブルタイトルをダブルクリックします。
   または
   テーブルプロパティを表示するインスペクターで**編集...**ボタンをクリックします。
   メソッドエディターウィンドウが表示されます。作成したばかりのトリガーは空です。

エクスプローラーからトリガーを作成/編集するには:

1. エクスプローラーのCurrent form tableを表示し、"トリガー"項目を展開します。
2. トリガーを作成するテーブルを選択し、リスト下部の追加ボタン をクリックします。
   または
   対象テーブルをダブルクリックします。
   トリガーがメソッドエディターのウィンドウに表示されます。

### フォームメソッドの作成/編集

フォームメソッドはエクスプローラーのCurrent form tableまたはフォームエディターから開くことができます。

エクスプローラーでフォームメソッドを作成するには:

1. エクスプローラーのCurrent form tableを表示します。
2. フォームメソッドを作成したいフォーム名をページ上に表示し、選択します。
3. リスト下部の追加ボタン をクリックします。
   または
   フォーム名をダブルクリックします。
   メソッドエディターウィンドウにフォームメソッドが表示されます。

**注:** エクスプローラーのフォームページからフォームメソッドを作成したり開いたりすることができます。フォーム名上でコンテキストメニューを表示し、**フォームメソッド編集...**を選択します。

フォームエディターからフォームメソッドを作成したり開いたりするには:

1. **フォーム**メニューから**フォームメソッド**を選択します。
   または
   フォームエディター上のオブジェクト以外の場所でコンテキストメニューを表示し、**フォームメソッド**を選択します。
   または
   プロパティリスト中でフォームメソッド行の**編集...**ボタンをクリックします。

メソッドエディターウィンドウにフォームメソッドが表示されます。

## メソッドの削除

データベースメソッド、トリガー、フォームメソッド、プロジェクトメソッドはエクスプローラーを使用していつでも削除できます。オブジェクトメソッドはフォームエディターで削除します。

#### プロジェクトメソッド、フォームメソッド、トリガー、データベースメソッドの削除

フォームメソッド、プロジェクトメソッド、トリガー、データベースメソッドはエクスプローラーから削除できます:

1. エクスプローラーでCurrent form tableを表示します。
   確実にメソッドページが開かれていることを確認してください。例えばフォームページが開かれていると誤ってフォームを削除してしまうかもしれません。
2. 削除するメソッドに対応する項目を選択します。
3. エクスプローラーウィンドウ下部にある削除ボタン をクリックします。

**注:** プロジェクトメソッドを削除するには、プロジェクトメソッド名上でコンテキストメニューを表示し、**メソッド削除**を選択することもできます。

- フォームメソッド、トリガー、データベースメソッドを削除する場合、操作の取り消しができない旨の警告が表示されます。
- プロジェクトメソッドを削除する場合は警告が表示されません。削除したメソッドはゴミ箱に移動され、ゴミ箱を空にしない限り元に戻すことが可能です (ゴミ箱ページ参照)。

#### オブジェクトメソッド削除

必要のなくなったオブジェクトメソッドはフォームエディターから削除できます。

オブジェクトメソッドを削除するには:

1. 削除したいメソッドが割り当てられたオブジェクトを含むフォームをフォームエディターで開きます。
2. オブジェクトを選択します。
3. **オブジェクト**メニューから**オブジェクトメソッド消去**を選択します。
   4Dは選択されたオブジェクトのオブジェクトメソッドを消去します。

**注:** 誤ってオブジェクトメソッドを削除してしまった場合、**編集**メニューから**取り消し**を選択します。

#### メソッドの書き出しと読み込み

すべてのタイプのメソッドをファイルに書き出し、またファイルから読み込むことが可能です。これを行うためのコマンドはメソッドエディターの**メソッド**メニューにあります。

- **メソッド書き出し...**コマンドを選択すると、標準のファイル保存ダイアログが表示され、書き出しファイルの名前と場所、およびフォーマットを選択することができます。メソッドブロックが折りたたまれているかどうかに関わらず、メソッド全体が書き出されます。
- **メソッド読み込み...**コマンドを選択すると、標準のファイルを開くダイアログが表示され、読み込むファイルを選択することができます。
  読み込みを行うと、メソッド中で現在選択されているテキストが置き換えられます。メソッド読み込みによって既存のメソッド全体を書き換えるには、読み込みを行う前に既存のメソッド全体を選択します。

読み込み/書き出し機能はマルチプラットフォーム対応です。Mac OSで書き出したメソッドをWindowsで読み込むことができ、またその逆も可能です。4Dは必要に応じて文字の変換を行います。

#### メソッドファイルフォーマット

メソッドの読み込みと書き出しに使用されるファイルは2つのフォーマットがあります:

- **4Dメソッド** (".c4d"拡張子): このフォーマットではメソッド中のオブジェクト名 (コマンド名など) がトークナイズされ、エンコードされます。
  このフォーマットは特に異なる言語 (非英語のコマンド名) を使用する4D間でメソッドをやり取りする場合に便利です。その代わりテキストエディターでメソッドを表示することはできません。
- **テキスト** (".txt"拡張子): このフォーマットではメソッドがテキスト形式で書き出されます。書き出したファイルは標準のテキストエディターで開くことができます。しかし書き出しと読み込みに使用される4Dアプリケーションの言語は同じでなければなりません (例えば英語のコマンド名を使用する4D)。

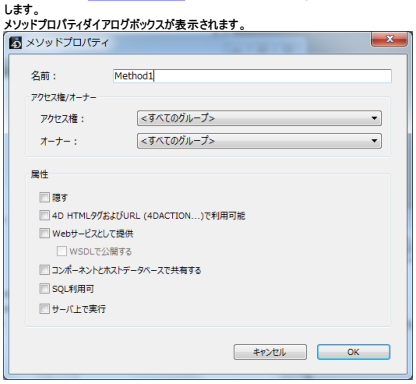
## プロジェクトメソッドプロパティ

プロジェクトメソッドを作成したら、名称やプロパティを変更できます。プロジェクトメソッドのプロパティは主に実行アクセスやセキュリティ条件 (ユーザーアクセス権やWebサーバー、Webサービスからのメソッド利用可否) に関する設定が含まれます。

他のタイプのメソッドには専用のプロパティがありません。これらのメソッドのプロパティは、それらが関連付けられているオブジェクトに基づいて決定されます。

プロジェクトメソッドのプロパティを変更するには:

1. WA SET URL FILTERSで**メソッド**メニューから**メソッドプロパティ...**を選択します。
   または
   エクスプローラーのCurrent form tableでプロジェクトメソッドを選択し、コンテキストメニューまたはオプションメニューから**メソッドプロパティ...**を選択します。
   メソッドプロパティダイアログボックスが表示されます。




**注:** 一回の処理で複数のプロジェクトメソッドの属性を設定するために、属性の一括設定を使用できます (属性一括設定参照)。

### 名前

メソッドプロパティウィンドウの"名前"エリアまたはエクスプローラーでプロジェクトメソッド名を変更できます。

新しい名前は4Dの命名規則に沿っていなければなりません (4Dランゲージリファレンスマニュアルの識別子参照)。同じ名前のメソッドが既に存在する場合、4Dはその旨を知らせるダイアログを表示します。名称を変更したら、4Dはメソッドリストをソートします。

**警告:** プロジェクトメソッドの名前を変更すると、そのメソッドを呼び出している他のメソッドやメニューの機能が無効になるリスクがあります。このため名称変更で説明されているメソッド名の名称変更機能を使用することを強く推奨します。この機能を使用すれば、データベース中でメソッド名が参照されている箇所がすべて自動的に変更されます (ただしEXECUTE METHODなど文字列としてメソッド名が参照される個所を除きます)。

4D Serverでは変更はサーバーに反映されます。複数のユーザーが同時に名称を変更しようとすると、最後に行われた変更の名称が適用されます。メソッドのオーナーを指定すれば特定のユーザーしかメソッド名を変更できないようにできます。

**注:** プロジェクトメソッド以外のメソッドの名称を変更することはできません。

### アクセス権/オーナー

ユーザーグループをアクセス権およびオーナーに設定して、メソッドへのアクセスを制御できます。ユーザーやグループの作成とその有効化についてはユーザー＆グループを参照してください。

- アクセス権ドロップダウンリストではそのメソッドを実行できるグループを設定します。指定されたグループに属さないユーザーがメソッドを実行しようとすると、4Dはユーザーにメソッド実行権限がない旨のメッセージを表示します。
- オーナードロップダウンリストではデザインモードでそのメソッドを編集できるグループを設定します。指定されたグループに属さないユーザーがメソッドの編集を試みると、4Dはユーザーにメソッド編集権限がない旨のメッセージを表示します。

両方のグループの属するユーザーは制限なしにそのメソッドを編集および実行できます。

### 属性

属性を使用して、プロジェクトメソッドがどの版で利用可能かを設定できます。エクスプローラーを使用して複数のメソッドに対し属性をいかっつして設定することもできます (後述)。

#### 非表示

特定のメソッドを、ユーザーに対し**実行**メニューの**メソッド...**から実行させたくない場合、このオプションを選択すればそのメソッドを非表示にできます。非表示のメソッドはメソッド実行ダイアログボックスに表示されなくなります (メソッド実行ダイアログボックスからメソッド実行参照)。

プロジェクトメソッドを非表示にしても、データベースプログラマーはそれを使用することができます。メソッドを非表示にしてもエクスプローラーのCurrent form tableやメソッドエディターのメソッドリストには表示されます。

#### 4D HTMLタグおよびURL (4DACTION...) で利用可能

このオプションは4D Webサーバーのセキュリティを強化するために使用されます。このオプションが選択されていない場合、そのメソッドは4DACTIONや4DMETHOD URLを使用したHTTPリクエスト、および4DSCRIPT/4DTEXT/4DHTMLタグ (さらには4DVARや4DHTMLVARタグ) から直接呼び出すことができません。詳細は4Dランゲージリファレンスマニュアルの接続セキュリティを参照してください。

エクスプローラー中ではこの属性が設定されたプロジェクトメソッドには特別なアイコンが表示されます 。

セキュリティのためこのオプションはデフォルトで選択されていません。Web機能から直接呼び出されるメソッドには明示的にこのオプションを選択しなければなりません。

#### Webサービスとして提供

この属性を使用してSOAPリクエストに応答するようメソッドを構成することができます。詳細はWeb サービスの公開と使用を参照してください。このオプションを選択すると、**WSDLで公開する**オプションを選択できるようになります。

エクスプローラーではWebサービスとして提供されるプロジェクトメソッドには以下のアイコンが表示されます 。

**注:** メソッド名がXMLの命名規則に準拠しない文字 (例えばスペース) を含む場合、そのメソッドをWebサービスとして公開することはできません。この場合、設定は保存できません。

#### WSDLで公開する

この属性は"Webサービスとして公開する"属性が設定されたメソッドでのみ設定できます。この属性を設定すると、そのメソッドが4Dアプリケーションの WSDLに含まれます。詳細はWSDL ファイルを生成するを参照してください。

エクスプローラーではWSDLで公開されたWebサービスメソッドには以下のアイコンが表示されます 。

#### コンポーネントとホストデータベースで共有する

この属性はコンポーネントのフレームワークで使用されます。このオプションが選択されていると、アプリケーションがホストデータベースとして実行されている場合、そのメソッドがコンポーネントから実行可能になります。またアプリケーションがコンポーネントとして実行されている場合、そのメソッドはホストデータベースから実行可能となります。

コンポーネントについては4Dコンポーネントの開発とインストールを参照してください。

#### SQL利用可

この属性が選択されていると、そのプロジェクトメソッドは4DのSQLエンジンから実行可能となります。デフォルトでは選択されておらず、明示的に許可されない限り、4Dメソッドを4D SQLエンジンから呼び出すことはできません。

このプロパティはすべての内部および外部SQLクエリ (ODBCドライバー経由、Begin SQL/End SQL タグ内のSQLコード、またはQUERY BY SQLからの呼び出し) に適用されます。

**注:**

- メソッドに"SQL利用可"属性が設定されていても、メソッドの実行時にはデータベース設定およびメソッドプロパティに設定されたアクセス権が考慮されます。
- ODBCの**SQLProcedure**関数はこの属性が設定されているプロジェクトメソッド名のみを返します。

詳細はSQLマニュアルのMissingRefを参照してください。

#### サーバー上で実行

この属性はクライアント/サーバーモードの4Dアプリケーションでのみ考慮されます。このオプションが選択されていると、そのプロジェクトメソッドは常にサーバー上で実行されます。

このオプションに関する詳細は4D Serverリファレンスマニュアルのサーバー上で実行属性を参照してください。

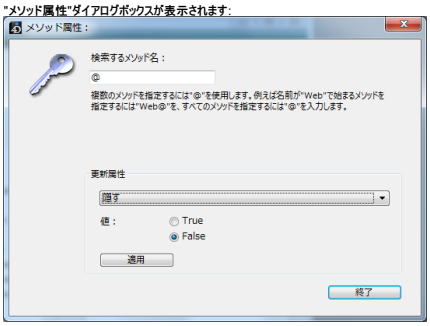
**属性一括設定**

"メソッド属性"ダイアログボックスを使用して、一回の操作で複数のプロジェクトメソッドに対して属性を設定することができます。この機能は多数のプロジェクトメソッドの属性を一括して設定する場合に便利です。

メソッド属性の一括設定を行うには:

1. エクスプローラーのCurrent form tableのツールメニューから**属性一括設定...**コマンドを選択します。

"メソッド属性"ダイアログボックスが表示されます:




2. "検索するメソッド名"エリアに属性を一括設定するメソッドを決定するための名前条件を入力します。
   入力した文字列を使用してメソッド名が検索されます。
   "@"をワイルドカード文字として使用し、メソッドグループを選択できます。すべてのメソッドを選択するには"@"のみを入力します。
   **注:**
   - 大文字と小文字は区別されません。
   - "@"は複数回使用できます (例: "dtro_@web@pro@")
3. "更新属性"エリアではドロップダウンリストから更新対象の属性を選択し、**True** または **False** ラジオボタンを選択します。
   **注:** "WSDLとして公開"属性をTrueに設定しても、この属性は"Webサービスとして公開"属性がTrueに設定されたメソッドにのみ適用されます。
4. **適用**をクリックします。
   属性変更は検索条件に合致するメソッドに対し、即座に反映されます。

# メソッドエディター

メソッドエディターウィンドウはいくつかのエリア+(ツールバー、編集エリア、リストエリア、ステータスバー、ブレークポイントエリア)+から構成されています。

リストエリアにリストを追加したり、行番号を隠したりするなど、メソッドエディターウィンドウの表示をカスタマイズできます。

## ツールバー

メソッドエディターにはメソッドの実行と編集に関連する基本的な機能に素早くアクセスするためのツールバーがあります。

| | |
|---|---|
| **メソッド実行** | このボタンをクリックするとメソッドが実行されます。ボタンに割り当てられているメニューを使用して実行タイプを選択できます。このボタンはプロジェクトメソッドと一部のデータベースメソッドでのみ有効です。詳細はメソッドの実行を参照してください。 |
| **検索** | 検索アイコンをクリックすると標準の検索ダイアログボックスが表示されます。このダイアログを使用してそのメソッド中を検索できます。ボタンの右にあるコンボボックスに検索キーワードを入力し**Enter** または **Carriage return**を押すと直接検索が行われます。またコンボボックスには検索履歴も表示されます。メニューから検索文字を指定して検索を実行することもできます。詳細はメソッド中の検索と置換を参照してください。 |
| **マクロ** | このボタンをクリックすると、利用可能なマクロがすべて表示されます。詳細はマクロの作成と利用を参照してください。 |
| **すべて折りたたむ / すべて展開する** | これらのボタンを使用してメソッドの制御フロー構造を折りたたんだり展開したりできます。詳細はメソッドを記述するを参照してください。 |
| **リストの表示/非表示** | このボタンを使用してウィンドウ下部のリストの表示/非表示を切り替えられます。詳細はこのページの"リストエリア"を参照してください。 |
| **メソッド情報** | このボタンをクリックするとメソッドプロパティウィンドウにアクセスできます (プロジェクトメソッドのみ)。詳細はプロジェクトメソッドプロパティを参照してください。 |
| **最新のクリップボード値** | このボタンには、ウィンドウ中でコピーされた最新の20個の文字列がメニューに表示されます。項目を選択するとその文字列がカーソルの位置に挿入されます。 |
| **クリップボード** | これら9つのアイコンは、メソッドエディターで利用可能な9つのクリップボードを表します。アイコンに番号が表示されている場合、クリップボードにデータが含まれていることを表します。オレンジアイコンのクリップボードは空です。これらのクリップボードはアイコンをクリックするかキーボードショートカットで利用できます。詳細はメソッドを記述するを参照してください。 |
| **ロック**(4D Serverのみ) | このアイコンが表示されている場合、そのメソッドは他のユーザーが開いているためロックされていることを表します。南京錠をクリックするとメソッドを再読み込みできます。(結果他のユーザーがhぞオンした内容が反映されます。) |

## 編集エリア

編集エリアにはメソッドテキストが含まれます。このエリアを使用してメソッドテキストを入力/編集します。エディターは自動でメソッドテキストのインデントを設定し、異なるシンタックスごとに色を付けます。

このエリアの表示はカスタマイズできます。カスタマイズ内容はすべてのメソッドエディタに反映されます。

- **フォント** と **フォントサイズ**: メソッドエディター編集エリアの文字フォントとサイズは4D環境設定のメソッドページで設定します。また**メソッド** > **表示**メニューから**フォントを大きくやフォントを小さく**サブメニューを選択して変更することもできます。
- **システム要素のスタイルとカラー**: 4DやSQLランゲージの各要素タイプごとに異なる色やスタイルを割り当てることができます。これを行うにはランゲージ要素 (変数やキーワード等) 上でコンテキストメニューを表示し、**スタイル**サブメニューからオプションを選択します。変更はすべてのエディターの同じタイプの要素に適用されます。このオプションは4D環境設定のメソッドページでも設定できます。
- **スペース**: 単語間のスペースを空白ではなくピリオドで表示することができます。これを行うには**メソッド**メニューから**表示 > ホワイトスペースと**選択します。この設定はすべてのコード要素に適用されます。

```
APPEND TO ARRAY($_CurLang;"en") APPEND·TO·ARRAY($_CurLang;"en")
```

- **コードインデント幅**: このオプションは4D環境設定のメソッドページで設定できます。
- **インターフェースカラー**: 編集エリアのインターフェース (ハイライト、背景等) に使用されるカラーを4D環境設定のメソッドページで設定できます。

エリアへのコード入力についてはメソッドを記述するを参照してください。

## リストエリア

リストエリアにはメソッドの記述に必要となるコード要素 (コマンド、定数、フォーム等) のリストを1つ以上表示できます。表示するリストの数や種類を変更できます。

### リスト表示

デフォルトで4つのリストが表示されます。リストの境界をドラッグして相対的にリストの幅を調整できます。またリストと編集エリアの境界線をドラッグするとリストの高さを調整できます。

- リスト中の項目をダブルクリックすると、現在のカーソル位置にその項目が挿入されます。
- **リストに表示内容を変更する**: リストのタイトルエリアをクリックするとポップアップメニューが表示され、その中から表示したい項目を選択します。
- **リストを追加/削除する**: リストのタイトルエリアをクリックし、一番下の対応するメニューを選択します。
  エディターウィンドウには最低一つのリストが表示できるよう設定されていなければなりません。**このリストを削除**コマンドは、残りリストが1つのときは無効になります。リストを表示したくない場合はツールバーの ボタンをクリックするか、環境設定でリストのデフォルト表示モードを設定します。
- すべてのウィンドウでリストを非表示にするには**メソッド**メニューの**表示** >**リスト**オプションの選択を解除するか、4D環境設定のメソッドページページで**リスト表示**オプションの選択を解除します。設定の変更は次回メソッドを開いた際に有効となります。

メソッドエディター中で設定した内容をテンプレートとして保存できます。詳細はテンプレートとして保存を参照してください。

### リストのタイプ

メソッドエディターウィンドウのリストエリアには以下の項目を表示できます:

- **すべてのテーブル/フィールド**: データベーステーブルとフィールドの名前 (階層リスト形式)。ダブルクリックでフィールド名を挿入すると、4Dはシンタックスに合わせてテーブル名も挿入します。
- **テーブル** (サブメニュー): サブメニューを使用して選択したテーブルのフィールド名。
- **カレントテーブル**: カレントテーブルのフィールド名 (トリガー、フォームメソッド、オブジェクトメソッドで利用可能)。
- **プロジェクトフォーム**: データベースに定義されたプロジェクトフォーム名。プロジェクトフォーム名をダブルクリックすると、4Dはダブルクォートで囲まれたフォーム名を挿入します。
- **テーブルフォーム**: データベースに定義されたテーブルフォーム名 (階層リスト形式)。ダブルクリックでテーブルフォーム名を挿入すると、4Dはテーブル名の後にセミコロン、そしてダブルクォートで囲まれたフォーム名を挿入します。
- **メソッド**: データベースに定義されたプロジェクトメソッド名。
- **すべてのフォルダー**: データベースに定義されたオブジェクトフォルダーおよびサブフォルダー名 (階層リスト形式)。フォルダーはカスタマイズされた方法でオブジェクトをグループ化するために使用します。フォルダーはエクスプローラーのホームページで管理します。
- **フォルダー** (サブメニュー): サブメニューを使用して選択されたフォルダーの中身。
- **マクロ**: データベースに定義されたマクロ名 (マクロの作成と利用参照)。
- **コマンド**: 4Dランゲージコマンド (文字順)。
- **コマンド (テーマ別)**: テーマごとにグループ化された4Dランゲージコマンド (階層リスト形式)。
- **メニューバー**: 4Dメニューバーエディターで作成したメニューバーの名前と番号 (メニューとメニューバー参照)。
- **定数**: 4Dやプラグインに定義された定数 (テーマ別階層リスト形式)。
- **リスト**: リスト名。
- **ピクチャー**: ピクチャーライブラリに格納されたピクチャーの名前と番号。
- **すべてのプラグインコマンド**: データベースにインストールされたプラグインコマンド名 (テーマ別階層リスト形式)。
- **プラグインコマンド** (サブメニュー): サブメニューを使用して選択されたプラグインのコマンド名。
- **SQLキーワード**: 4DのSQLシンタックスパーサーが解釈可能なキーワードのリスト。このリストにはコマンド、句、関数が含まれます。
- **SQL関数**: 4D SQL関数。

**注:** マクロ要素を除き、すべてのリストは文字順です。

## ブレークポイントエリア

編集エリアの左にあるこのエリアには行番号を表示したり、ブレークポイントを直接挿入したりできます。ブレークポイントはプログラム開発フェーズのデバッグに使用します。プログラムはブレークポイントが設定された場所で実行を停止し、デバッガーを表示します (インタープリターモードのみ)。

ブレークポイントを挿入するには、ブレークポイントを挿入したい行のブレークポイントエリアをクリックするか、その行で表示したコンテキストメニューの**ブレークポイントをトグル**コマンドを選択します。ブレークポイントが挿入された行には赤点が表示されます。

```
15    If (Count parameters>=1)
16 ●   FORM SET INPUT($1->;"INPUT FORM")
17    End if
18    End if
```

データベースに設定されたブレークポイントのリストはランタイムエクスプローラーで確認できます。

ブレークポイントを削除するには赤点をクリックするか、**ブレークポイントをトグル**コマンドを再度選択します。

一時的にブレークポイントを無効にしたりそのプロパティを変更するには、**Alt** キー (Windows) または **Option** キー (Mac OS) を押しながらブレークポ

イントをクリックするか、エディターのコンテキストメニューから**ブレークポイント編集**コマンドを選択します。ブレークポイントに関する詳細な説明は4Dランゲージリファレンスのブレークポイントを参照してください。

#### 行番号の表示/非表示

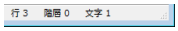
メソッドエディターの各ウィンドウごとにブレークポイントエリアの行番号表示/非表示を切り替えられます。

- 行番号表示/非表示のデフォルト設定は4D環境設定メソッドページの**行番号表示**で設定します。
- 各ウィンドウの設定を切り替えるには、メソッドメニューの**表示** > **行番号**コマンドを選択します。

行番号を表示すると、ウィンドウ中で特定の行を見つけやすくなります。**メソッド**メニューの**行番号を指定して移動...**コマンドを利用する場合も、この行番号を使用します (メソッド中の検索と置換の"行番号を指定して移動"参照)。

### ステータスバー

エディターウィンドウの右下にはステータスバーがあり、カーソルの位置を表示します:




- **行**: 行番号
- **階層**: 階層番号。プログラム構造の階層レベル。先頭のレベルは0です。この数値はコード中でエラーが発生した際にインタープリターから提供されるのでデバッグ時に利用できます。
- **文字**: 行中の文字位置。

### テンプレートとして保存

メソッドエディターウィンドウに設定したパラメーターをテンプレートとして保存できます。テンプレートとして保存すると、新規メソッドエディターを開くたびにその設定が使用されます。

テンプレートには以下のパラーメーターが保存されます:

- 編集とリストエリアの相対サイズ
- リストの数
- 各リストの位置と表示内容
- 各リストの相対幅

設定をテンプレートとして保存するには、**メソッド**メニューから**テンプレートとして保存**を選択します。テンプレートは4Dアプリケーションの環境設定に即座に保存されます。既存のテンプレートは上書きされます。

## メソッドを記述する

4Dのメソッドエディターはテキストエディターのように動作します。メソッドの記述はテキストの入力、コンポーネントの選択、そしてエクスプローラーやその他のウィンドウからのドラッグ操作により行います。メソッドの記述をより効率的に行うためにタイプアヘッド機能も提供されています。

1つのメソッドには最大32,000行、2GBのコードを記述できます。

4Dのメソッドエディターは基本的なシンタックスチェック機能を提供します。メソッド実行時には追加のエラーチェックが行われます。

### テキストの入力と編集

メソッドエディターへの入力と編集には標準のテキスト編集テクニックを使用します。入力を行うと挿入ポイントに文字が表示されます。**Return** または **Enter** キーを押すと改行が挿入されます。

**注:** テンキーの**Enter**キーとメインキーボードの**Enter**キーでは押したときの動作が異なります。テンキーの**Enter**キーを押した場合、4Dは改行を挿入せずに、その行のシンタックスチェックを行います。

メソッドエディターはシンタックス要素ごとに、指定されたスタイルやカラーを使用して表示を行います。この設定は変更できます (メソッドエディター参照)。入力を行う間、また改行を挿入する際、4Dは行のテキストを評価して適切な表示フォーマットを適用します。また{}などのプログラム構造が使用された場合、4Dは自動でインデントを行います。

矢印キーを使用して異なる行に移動できます。矢印による移動では行の評価が行われないため、クリックよりもすばやく移動できます。

Windowsでは日本語や中国語を入力するためにIMEがサポートされています。

メソッドエディターではナビゲーションショートカットがサポートされています。これらのショートカットについてはこのページで後述します。

#### ドラッグ＆ドロップによるメソッドオブジェクトの追加

メソッドを記述する際にドラッグ＆ドロップメカニズムを使用できます。エクスプローラー、同じメソッド内、2つのメソッド間で項目をドラッグ＆ドロップできます。

- エクスプローラーからは以下のものをドラッグ＆ドロップできます:
  - ホームページからテーブル名、フィールド名、フォーム名、そしてプロジェクトメソッド。
  - テーブルページからテーブル名とフィールド名。
  - フォームページからテーブル名とフォーム名。
  - メソッドページからメソッド名。
  - 定数ページから定数。
  - コマンドページから4Dコマンド。

  **注:** 特定のオブジェクト (特に定数やコマンド) の場合、ドラッグ＆ドロップをする際に**Ctrl** (Windows) または **Command** (Mac OS) キーを押さなければなりません。
  ドラッグ＆ドロップを行うと、4Dは常に正しいシンタックスでオブジェクトを挿入します。例えば[People]テーブルのFirstNameフィールドを挿入すると、メソッドエディターには"[People]FirstName"が表示されます。同様に[People]テーブルの"Input"フォームをドラッグ＆ドロップすると、[People];"Input"が挿入されます。
  (**Ctrl** (Windows) または **Command** (Mac OS) キーを押しながら) エクスプローラーのコマンドページからコマンドをドラッグすると、メソッドエディターにはシンタックスも挿入されます。このシンタックスは必要に合わせて編集しなければなりません。この機能はコマンドが期待する引数を確認するために存在します。

- メソッド内あるいは2つのメソッド間でのドラッグ＆ドロップ:
  テキストの一部を選択すると、メソッドエディターでのドラッグ＆ドロップメカニズムが有効になります。
  デフォルトではドラッグ＆ドロップによりテキストが移動されます。コピーを行うには**Ctrl** キー (Windows) または **Option** キー (Mac OS) を押しながら操作します。

#### 行を分割する

1つのステートメントを複数行に記述できるようになりました。これを行うにはステートメントの各行をバックスラッシュ "\" 文字 (日本語Windowsの場合半角円マーク) で終了します。4Dはこの文字で終わらない行までを一行として扱います。例えば以下のステートメントは同等です:

```
[DOCS]Plural:=Uppercase([DOCS]Plural[[1]])+Lowercase(Substring([DOCS]Plural;2))
```

```
[DOCS]Plural:=Uppercase([DOCS]Plural[[1]])+\
Lowercase(Substring([DOCS]Plural;2))
```

#### 文字種の変換

**注**: この節の内容はアルファベットに対してのみ有効です。

**メソッド**メニューの**大文字/小文字**サブメニュー、またはエディターのコンテキストメニューを使用して、選択したテキストの大文字/小文字を変更することができます:

- **大文字** / **小文字**: 選択した文字を大文字または小文字に変換します。
- **キャメルケース (先頭小文字)** / **キャメルケース (先頭大文字)** : 選択した文字をキャメルケースに変換します。
  テキストにこれらのコマンドを適用すると、スペースと"_"文字は取り除かれ、各単語の先頭文字が大文字になります。

#### 式の入れ替え

**式の入れ替え**機能を使用すると、代入式の左辺と右辺が入れ替わります。例えば:

```
variable1:=variable2
```

は以下のようになります。

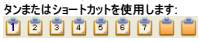
```
variable2:=variable1
```

この機能は特にプロパティを取得したり設定したりする一連の代入式を逆にする場合に便利です。この機能を使用するには、対象の行を選択し、**メソッド**メニューまたはエリアのコンテキストメニューから**式の入れ替え**コマンドを選択します。選択された行のうち、値を代入する行だけが更新されます。

#### コピー/ペースト文字列を複数記憶する

標準のコピー/ペースト処理に加え、4Dは異なるクリップボードの内容に対して動作する2つの追加の機能を提供します:

- 4Dは現在のセッション中メソッドエディター中で行われた最新の10個のコピーやカットアクションをメモリに格納します。格納されたこれらの内容はいつでも再利用できます。これを行うにはメソッドエディターコンテキストメニューの**クリップボード履歴**コマンドを使用するか、ツールバーのクリップボード履歴ボタンを使用します:

  
  コピー/カットされたテキストの最初の数文字が表示されます。項目を選択するとカーソル位置に、保存されていたテキストが挿入されます。
- そのほか、9つの番号付けされたクリップボードを利用することもできます。これらのクリップボードにアクセスするにはメソッドエディターツールバーのボタンまたはショートカットを使用します:
  

| | 選択したテキストをクリップボードにコピー | カーソル位置にクリップボードの内容をペースト |
|---|---|---|
| **Windows** | クリップボードアイコン上でCtrl+Alt+クリック | クリップボードアイコン上でCtrl+クリック |
| | Ctrl+Alt+クリップボード番号 | Ctrl+クリップボード番号 |
| **Mac OS** | クリップボードアイコン上でCmd+Option+クリック | クリップボードアイコン上でCmd+クリック |
| | Cmd+Option+クリップボード番号 | Cmd+クリップボード番号 |

#### 行の移動

カーソルのある行を、**メソッド**メニューの**行を上に移動**/**行を下に移動**コマンド、または**Alt/Option** + **上矢印**/**下矢印**で上下に移動することができます。行全体を選択する必要はありません。

#### 更新行の通知

メソッドをエディターで開いた後に更新された行には色つきのバーが表示されます:

```
14    ALL RECORDS([DOCUMENTATIONS])
15    ARRAY TEXT(<>KeyList;0)
16    ARRAY TEXT(<>PluralKeyList;0)
```

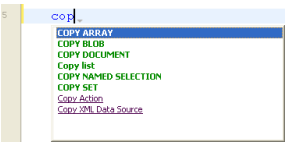
- 黄色: 行が変更された後に保存されていません。
- 緑色: 行が変更された後に保存されました。

### 自動補完機能

メソッドエディターには自動補完機能があります。4Dはタイプされた最初の文字に基づき、自動で候補を提案します。

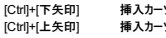
以下の例題の場合、"cop"とタイプしたことにより三角が表示されています。これは複数の候補があることを示しています:

**Tab**キーを押すと、候補がポップアップリストに表示されます:

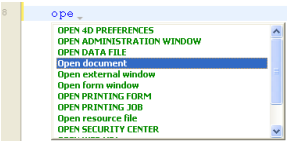



リストは要素のタイプごとにグループ化され、そのグループ内で文字順になっています。ダブルクリックまたは矢印キーで選択してから**Enter**, **Carriage Return** または **Tab**キーを押すと、選択された値が挿入されます。

**注:** 1つしか候補がない場合、その候補が灰色の文字で直接表示されます。**Tab**キーを押すとその候補が挿入されます:




デフォルトでは以下の区切り文字を入力すると、自動で候補も (区切り文字の前に) 挿入されます **( ; : = < [ {** 。




+ **(** キー -->

**注:** この区切り文字に伴って候補が自動で入力される動作を、環境設定のメソッドページで無効にできます。

**Esc** キーを押すとポップアップが閉じられます。またポップアップが開いた状態で入力を継続することもできます。文字が入力されるに伴い、ポップアップで提案される内容も更新されます。

タイプされた名前が異なる複数のオブジェクトに対応する場合、オブジェクトは対応するスタイルを使用してリスト中に表示されます。以下のタイプのオブジェクトが表示されます:

- 4D コマンド
- SQL コマンド
- プロジェクトメソッド
- テーブル名
- フィールド名
- 定数
- メソッド中で宣言されているローカル、プロセス、インタープロセス変数
- プラグインコマンド
- 4D キーワード
- SQL キーワード
- マクロ (<と>の間に表示)

**注: 定数**、**ローカル**および**インタープロセス変数**、**テーブル**に対する自動的なリストの表示を無効にできます。これらのオプションは環境設定のメソッドページにあります。

### テキストの選択

#### ブロックを選択

**ブロックを選択** 機能を使用して、挿入カーソルが含まれるコードブロックを選択できます。コードブロックは以下のように決定されます:

- クォーテーションマーク
- 括弧
- 論理構造 (If/Else/End if, While/End while, Repeat/Until Case of/End case)
- 中括弧

すでにブロックが選択されている状態でこの機能を使用すると、さらに上のレベルのブロックを選択します。

**Ctrl+Shift+B** (Windows) または **Command+Shift+B** (Mac OS) を使用するとこの操作を逆に行うことができ、最後に選択されたブロックの選択が解除されます。

**注:** 挿入ポイントがIf または Else タイプの構造の中にある場合、If または Else 構造ブロックが選択されます。

#### ダブルクリック

単語を選択するには、選択したい単語上でダブルクリックを行います。参照される名前 (コマンド、定数、メソッド等) にスペースが含まれる場合、**Alt/Option + ダブルクリック**を行うとスペースを含む名前全体を選択できます。

#### ナビゲーションキーボードショートカット

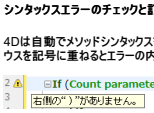
メソッドエディターでコード中を移動するために、標準のキーボードショートカットを使用できます。

**注:** Mac OSでは**Ctrl**を**Command**に読み替えてください。

| ショートカット | アクション |
|---|---|
| [Shift]+[右矢印] | 選択する範囲を一文字ずつ右方向に移動します。 |
| [Shift]+[左矢印] | 選択する範囲を一文字ずつ左方向に移動します。 |
| [Shift]+[下矢印] | 選択する範囲を一文字ずつ下方向に移動します。 |
| [Shift]+[上矢印] | 選択する範囲を一文字ずつ上方向に移動します。 |
| [Ctrl]+[Shift]+[右矢印] | 選択する範囲を単語単位で右方向に移動します。 |
| [Ctrl]+[Shift]+[左矢印] | 選択する範囲を単語単位で左方向に移動します。 |
| [Ctrl]+[下矢印] | 挿入カーソルを単語単位で左から右に移動します。 |
| [Ctrl]+[上矢印] | 挿入カーソルを単語単位で右から左に移動します。 |
| [Home] | 挿入カーソルを行の先頭に移動します。 |
| [End] | 挿入カーソルを行の最後に移動します。 |
| [Ctrl]+[Home] | 挿入カーソルをメソッドの先頭に移動します。 |
| [Ctrl]+[End] | 挿入カーソルをメソッドの最後に移動します。 |
| [Shift]+[Home] | 行の先頭からカーソルまでを選択します。 |
| [Shift]+[End] | カーソルから行の最後までを選択します。 |
| [Page Up] | メソッドをページ単位で上にスクロールします (挿入カーソル位置は変更されません)。 |
| [Page Down] | メソッドをページ単位で下にスクロールします (挿入カーソル位置は変更されません)。 |

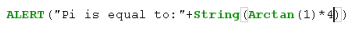
### シンタックスエラーのチェックと訂正

4Dは自動でメソッドシンタックスをチェックします。シンタックスが正しくない状態では、そのことを示す記号がブレークポイントエリアに表示されます 。マウスを記号に重ねるとエラーの内容が表示されます:




コードを入力する際、テンキー側のEnterを押すと、改行されることなく現在の行のシンタックスチェックを行えます。4Dは行を評価し、フォーマットし、エラーがあればそれを表示し、挿入カーソルを行の最後に移動します。メソッド行にエラーがあると表示された場合はそれを修正してください。正しく修正されると4Dはエラー記号を取り除きます。メソッドを保存したりウィンドウを閉じたり、**Enter**キーを押したりしたときはメソッド全体が評価されます。

メソッドが評価されるとき、4Dは基本的なシンタックスエラーの他、If ... End if などの構造も検証します。また括弧やクォーテーションマークなどの対応 (ペアになっているか) もチェックします。閉じる文字を入力すると、4Dは開始と終了文字を灰色の四角で囲み、それらがペアであることを示します:



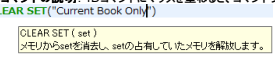

**注:** コード中のこれらの文字をクリックすると、4Dはデフォルトで対応するペアの文字を灰色四角で囲みます。この振る舞いは環境設定のメソッドページ中"括弧のマッチング"オプションで変更できます。

メソッドエディターではスペルミスなどの大きなシンタックスエラーしかチェックできません。実行時エラーはチェックできません。実行時エラーはメソッドが実行されているときに4Dが検出します。4Dはこのようなエラーを処理するためにデバッガー (デバッグ参照) を提供します。またコンパイラーを使用すれば型の非互換などのエラーを検出することもできます。コンパイラーについてはコンパイルを参照してください。

### ヘルプTipを使用する

メソッドエディターではカーソルの位置に応じて様々なヘルプTipが表示されます。

- **エラー**: ブレークポイントエリアにエラーアイコンが表示されているとき、マウスをアイコンに重ねるとエラーの内容が表示されます。
- **4Dコマンドの説明**: 4Dコマンドにマウスを重ねると、コマンドシンタックスと簡単な説明が表示されます。



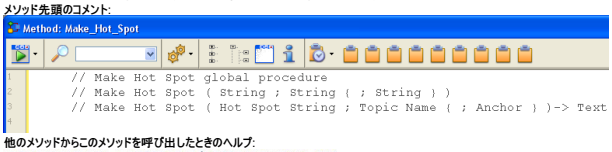
- **変数型**: 変数にマウスを重ねると、その変数の方が表示されます (そのメソッド中で宣言されている場合)。

```
If (<>vbWeStop)
  $vlcmd:=$vlNbCmd+1
End if
```

$vlNbCmd : C_LONGINT

- **プロジェクトメソッド**: プロジェクトメソッド呼び出しにマウスを重ねると、以下の内容が表示されます:
  - エクスプローラーで設定された、メソッドのコメント (コメントを使用する参照)、または
  - メソッド先頭のコメント (//で始まる行)。メソッドの先頭にコメント形式でそのメソッドや引数の説明を記述するのは一般的な習慣です。エクスプローラーコメントを削除すれば、こちらの内容をヘルプとして表示できます。

    メソッド先頭のコメント:




    他のメソッドからこのメソッドを呼び出したときのヘルプ:

### コマンドの完全なドキュメントを表示する

4Dランゲージコマンドの HTMLドキュメントを表示できます。これを行うには、コマンド名の途中にカーソルを挿入し、**F1**を押します。デフォルトブラウザーの新しいウィンドウにコマンドの完全なドキュメントが表示されます。

アプリケーション環境設定 (ドキュメントの場所) に基づき、4Dはオンラインドキュメントセンター、あるいはローカルからそのコマンドのドキュメントを探します。

### コメント/アンコメント

コメントを使用するとコード行を無効にできます。プログラムはコメント行をコードとして解釈せず (4Dはメソッドエディター上でコメントにスタイルを適用しません) メソッドを呼び出してもそのコードは実行されません。

コメントを作成するには、そのコメントの先頭に"//"を挿入します。

コメントの最大長は行と同じで32,000文字です。

**メソッド**メニューやメソッドエディターコンテキストメニューにある**コメント/アンコメント**コマンドを使用すると、選択した (複数の) 行を一度にコメント化したり、あるいは逆にコメント解除できます。
このコマンドを使用するには、コマンドを適用したい行を選択し、**コメント/アンコメント**を選択します:

```
If (Count parameters>=2)
  EXECUTE FORMULA(Command name(55)+"(['
Else
  If (Count parameters>=1)
    FORM SET INPUT($1->;"INPUT FORM")
  End if
End if
```

->

```
If (Count parameters>=2)
  EXECUTE FORMULA(Command name(55)+"(["+
    //Else
    //If (Count parameters>=1)
    //FORM SET INPUT($1->;"INPUT FORM")
    //End if
End if
```

選択行がアクティブなコードの場合、**コメント**コマンドが適用されます。選択行にアクティブ行と非アクティブ行が含まれる場合、すべての行に"//"が追加されます。この動作により、複数回コマンドを適用しても、最初のコメントレベルが維持されます。選択行がすべて非アクティブコードの場合、**アンコメント**コマンドが適用されます。

**注: コメント/アンコメント** コマンドは行全体に適用されます。行の一部をコメント化するためには使用できません。

### エスケープシーケンス

メソッドエディターでエスケープシーケンス (エスケープ文字) を使用できます。エスケープシーケンスは特別な文字の代わりに使用される一連の文字です。

エスケープシーケンスは一文字目がバックスラッシュであり、そのあとに文字が続きます。例えば "\t" は**タブ**文字のエスケープシーケンスです。エスケープシーケンスを使用すると特別な文字の入力が容易になります。例えば "\t" を **Char**(Tab) の代わりに使用できます。

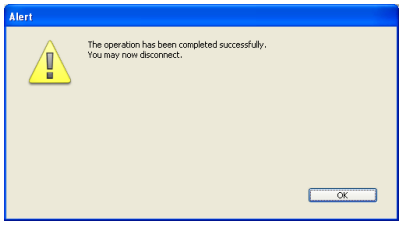
4Dでは以下のエスケープシーケンスを使用できます:

| エスケープシーケンス | 意味する文字 |
|---|---|
| \n | LF (行送り) |
| \t | HT (タブ) |
| \r | CR (改行) |
| \\ | \ (バックスラッシュ) |
| \" | " (引用符) |

**注:** エスケープシーケンスでは大文字小文字が区別されません。

以下の例題で**改行**文字 (エスケープシーケンス "**\r**") を使用して、警告ダイアログに表示されるメッセージに改行を挿入しています:

```
ALERT("The operation has been completed successfully.\rYou may now disconnect.")
```




**警告:** "\" (バックスラッシュ) はWindowsでパス名の区切り文字として使用されています。通常4Dはメソッドエディターに入力されたバックスラッシュを自動で"\\"に置き換えることで、これを正しく解釈します。例えば"C:\Folder"と入力すると"C:\\Folder"に変換されます。しかし"C:\MyDocuments\New"と入力した場合4Dは"C:\\MyDocuments\New"を表示します。この場合、二番目のバックスラッシュは "\N" (行送り) と解釈されてしまいます。このようなケースでは開発者がバックスラッシュを2つ入力するようにしなければなりません。
さらに正規表現のパターン定義でもバックスラッシュがエスケープシーケンスとして使用されます。正規表現パターン "\\" を4Dのメソッドエディターに記述する場合は "\\\\" となる点に注意してください。

### メソッドの編集を容易にする

メソッドエディターにはコードの可読性を高めたり、行のブラウズを容易にしたりする機能があります。

#### 折りたたむ/展開する

ループや条件分の中にあるコードを折りたたんだり展開したりして、メソッドの可読性を高めることができます:

- 展開されたコード:

```
If (Count parameters>=2)
  EXECUTE FORMULA(Command name(55)+"(["+$2+"];"INPUT FORM")")
Else
  If (Count parameters>=1)
    FORM SET INPUT($1->;"INPUT FORM")
  End if
End if
```

- 折りたたまれたコード:

```
If (Count parameters>=2)
  EXECUTE FORMULA(Command name(55)+"(["+$2+"];"INPUT FORM")")
Else
  If (Count parameters>=1)   ...
End if
```

展開ボタン [...] の上にマウスを重ねるとヘルプチップが表示され、非表示になっているコードの先頭部分を見ることができます。

折りたたまれたコード部分を選択、コピー、ペースト、または削除することができます。折りたたみに含まれるすべての行が操作の対象となります。このよ

うなコードがペーストされると、自動で展開されます。

コードを折りたたみ、展開する方法はいくつかあります:

- 折りたたみ/展開アイコンまたは (Windowsでは[+]と[-]、Mac OSでは三角) または閉じられた状態では開くボタン [...] をクリックする
- **メソッド**>**折りたたみ/展開**メニューを使用する:
  - **選択部を折りたたむ** / **選択部を展開する**: 選択されたテキスト中にあるすべてのコードを折りたたむまたは展開します。
  - **現在のレベルを折りたたむ** / **現在のレベルを展開する**: カーソルが存在するレベルのコードを折りたたむまたは展開します。このコマンドはメソッドエディターのコンテキストメニューからもアクセスできます。
  - **すべてを折りたたむ** / **すべてを展開する**: メソッド中のすべてのループと条件ブロックを折りたたむまたは展開します。このコマンドはメソッドエディターのツールバーからもアクセスできます: .

#### ブロックの先頭/ブロックの終端

コード構造の中を移動することを容易にするための機能があります。

- **ブロックの先頭**: カーソルを現在のブロックの先頭に移動します。
- **ブロックの終端**: カーソルを現在のブロックの終端に移動します。

このコマンドにはメソッドメニューとメソッドエディターのコンテキストメニューからアクセスできます。以下のショートカットも利用できます:

- Windows: **Ctrl + up arrow** または **Ctrl** + **down arrow**,
- Mac OS: **Command** + **up arrow** または **Command** +**down arrow**.

#### ブックマークを使用する

メソッドの特定の行にブックマークを付けることができます。ブックマークを付けると、特別なコマンドを使用してブックマーク間を素早く移動できます。

```
15
16    ARRAY TEXT ($_CurLang;0)
17
```

コードが挿入されたり削除されたりしてブックマーク行が移動すると、ブックマークもそれに伴って移動します。ブックマークはメソッドとともに保存されます。

ブックマークは**メソッド**メニューの**ブックマーク**サブメニューから管理します:

- **切り替える**: カーソルが存在するカレントの行にブックマークを設定します (まだブックマークが設定されていない場合)。すでにブックマークが設定されていればそれを取り除きます。このコマンドにはエディターコンテキストメニューの**ブックマークをトグル**、または**Ctrl+F3** (Windows) / **Command+F3** (Mac OS) キーボードショートカットからもアクセスできます。
- **すべてを削除**: 最前面のメソッドエディターからすべてのブックマークを取り除きます。
- **次に進む** / **前に戻る**: ウィンドウ中でブックマーク間を移動するために使用します。コマンドを選択すると、ブックマークが設定された行の先頭にカーソルが移動します。**F3** (次に進む) や **Shift+F3** (前に戻る) ショートカットも使用できます。

**注:** 検索で見つかった項目を含む行のマーカーとしてブックマークを使用できます。この場合4Dは自動でブックマークを追加します。詳細はメソッド中の検索と置換を参照してください。

# マクロの作成と利用

メソッド内でマクロコマンドを利用することができます。マクロを利用すればメソッドの入力にかかる時間を大幅に短縮することができます。

## マクロとは

マクロコマンドは4Dコードのパーツで、どのデータベースが開かれているかにかかわらず、あらゆるメソッドのどんな場所にでも挿入することができます。マクロにはすべてのタイプの4Dテキスト、コマンド、定数、さらにマクロ実行時のメソッドコンテキストにより値が置換される特別なタグを記述することができます。例えばマクロに<method_name/>というタグを記述しておくと、このタグはマクロ挿入時にカレンとのプロジェクトメソッド名に置換されます。

マクロは複数のXMLフォーマットのテキストファイルに格納されます。マクロはメソッドエディターリスト、またはエディター内のコンテキストメニュー、さらに自動補完機能から呼び出すことが可能です。

4DマクロはXMLフォーマットで記述されます。4Dが提供するマクロファイルをそのまま使用することも、変更することもできます。

## マクロの場所

4Dは"**Macros v2**"という名前のフォルダーからマクロをロードします。マクロはXMLファイルとしてこのフォルダーに配置しなければなりません。マクロファイルはフォルダー内に複数置くことができます。

"Macros v2"フォルダーは以下の場所に配置できます:

- マシンのアクティブな4Dフォルダー。マクロはすべてのデータベースで共有されます。
  **注:** アクティブな4Dフォルダーの位置はOSにより異なります。詳細は4Dランゲージリファレンスマニュアルの[Get 4D folder]コマンドの説明を参照してください。
- データベーストラクチャーファイルと同階層。マクロはこのストラクチャーにのみロードされます。
- コンポーネントの場合: コンポーネントのストラクチャーと同階層。この場合マクロはコンポーネントがインストールされている場合のみ利用できます。

これら3つの場所は同時に利用可能です。各場所に"Macros v2"フォルダーをインストールできます。マクロは以下の順番でロードされます: 4Dフォルダー、ストラクチャーファイル、コンポーネント。

## デフォルトマクロ

4Dはデフォルトのマクロを提供しています。このマクロは4Dを最初に起動する際に、マシンのアクティブな4Dフォルダー内の"Macros v2"フォルダーに、"*Macros.xml*"という名前で作成されるXMLファイル内に含まれます。

このファイルを変更することができます。またフォルダー内に別のマクロファイルを追加することもできます。このフォルダーに問題が発生した場合は、フォルダーを削除することで次回の4D起動時に新たに作成されます。

## カスタマイズしたマクロを追加する

標準のテキストエディターやプログラムを使用して、"Macros.xml"ファイルにカスタマイズしたマクロを追加できます。またフォルダー内にカスタマイズしたマクロを記述したファイルを追加することもできます。

ローカルモードでは4Dを使用中にマクロファイルを開くことができます。利用可能なマクロのリストは4Dがアクティブになるたびに更新されます。例えばテキストエディターを最前面にしてマクロファイルを更新し、その後メソッドエディターに戻ると、新しいマクロがメソッドエディターで利用可能になっています。

空およびエラーのあるマクロは表示されません。

### カスタマイズしたマクロのシンタックスを検証する

4DのマクロコマンドファイルはXML標準に準拠していなければなりません。つまりXML宣言 <?xml version="1.0" ...?> とドキュメント宣言 <!DOCTYPE macros SYSTEM "http://www.4d.com/dtd/2007/macros.dtd"> 文がマクロファイルの先頭になければならず、そうでない場合、マクロはロードされません。異なるタイプのXMLエンコーディングがサポートされていますが、Mac/PC (UTF-8) 互換のエンコーディングを使用することをお勧めします。4Dはマクロファイルを検証するためのDTDを提供しています。ファイルは以下の場所にあります:

- Windows: \Resources\DTD\macros.dtd
- Mac OS: :Contents:Resources:DTD:macros.dtd

マクロファイルに宣言文が含まれない場合や検証に失敗する場合、マクロはロードされません。

## 4Dマクロのシンタックス

4Dマクロは"要素"と呼ばれるカスタマイズされたXMLタグを使用して構築されます。

いくつかのタグは定義の開始と終了を表し (<tag>と</tag>のようなペア)、その他のタグは実行コンテキストにより置換されます (<tag/>)。

XML仕様に準拠し、いくつかのタグは属性を含むことがあります。明示されない限り属性の指定は任意であり、省略された場合はデフォルト値が使用されます。要素および属性のシンタックスは以下の通りです:

- 開始および終了タグのペア: <tag attribute="value"> </tag>
- 置換されるタグ: <tag attribute="value"/>

要素に複数の属性を指定できる場合、スペースで区切って記述します:
<tag attribute1="value" attribute2="value" attribute3="value"... >

利用可能なタグのリストと利用モードは以下の通りです:

| 要素タグ | 説明 |
|---|---|
| <macros> </macros> | マクロファイルの開始と終了 (必須)。 |
| <macro> </macro> | マクロ定義の開始と終了。<br>*属性*:<br>- name: メニューやメソッドエディターリストに表示されるマクロの名前** (必須)。<br>- type_ahead_text: タイプアヘッド (自動補完) 機能*を使用してマクロを呼び出すために入力する文字列**。<br>- in_menu: コンテキストメニュー*からマクロを呼び出すことを許可するかを指定するブール値。値は"true" (デフォルト) または "false"。<br>- type_ahead: タイプアヘッド (自動補完) 機能*を使用してマクロを呼び出すことを許可するかを指定するブール値。値は"true" (デフォルト) または "false"。<br>- method_event: メソッドごとの現在の処理フェーズ (作成、閉じる等) に応じてマクロの自動呼出しを実行するために使用します。値 = "on_load": マクロはメソッドを開くたびに実行されます, "on_save": マクロはメソッドが保存されるたびに実行されます (メソッドを閉じるまたはファイル>保存コマンドを使用して保存する), "on_create": マクロはメソッドが作成されるたびに実行されます, "on_close": マクロはメソッドが閉じられるたびに実行されます。<br>"on_save" と "on_close" は組み合わせて使用できます — 言い換えれば、これらのイベントは更新されたメソッドが閉じられるときに両方とも生成されます。他方、"on_create" と "on_load" が連続して実行されることはありません。この属性は例えばメソッド作成時にデフォルトのフォーマット (ヘッダーエリアのコメント等) を適用したり、メソッドを開いた際のタイムスタンプを記録するために使用することができます。<br>- version: マクロ用にテキスト選択の新しいモードを有効化するために使用します (後述の"<method>タグについて"の節参照)。新しいモードを有効にするには"2"を指定します。この属性を省略するかversion="1"を指定すると、以前のモードが保持されます。<br>- in_toolbar: ツールバーのマクロボタンメニューにマクロを表示するかどうかを指定するブール値。値="true" (デフォルト) または "false". |
| <selection/> | マクロが挿入される時点で選択されていたテキストを使用して、タグが置換されます。選択テキストは空の場合もあります。 |
| <text> </text> | メソッドに挿入されるコードの開始と終了タグ。コードの先頭と最後には改行が追加されます。 |
| <method> </method> | プロジェクトメソッド名とその引数 (オプション) の開始と終了。マクロが呼び出されたとき、メソッドが実行されます。("param1;param2;...")の形式で引数を渡せます。この引数は呼び出されたメソッド内で$1, $2...変数に渡されます。このタグに関する追加情報は後述の"<method>タグについて"を参照してください。 |
| <caret/> | マクロが挿入された後のコード中の挿入ポイントの場所。 |
| <user_4D/> | カレントの4Dユーザー名で置換されるタグ。 |
| <user_os/> | カレントのシステムユーザー名で置換されるタグ。 |
| <method_name/> | カレントのプロジェクトメソッド名で置換されるタグ。 |
| <method_path/> | カレントプロジェクトメソッドのフルパス名で置換されるタグ。 |
| <date/> | カレントの日付で置換されるタグ。<br>*属性*:<br>- format: 4Dの日付フォーマット。フォーマットが設定されていない場合、デフォルトのフォーマットが使用されます。値 = 4Dフォーマットの数字 (0 - 8). |
| <time/> | カレントの時間で置換されるタグ。<br>*属性*:<br>- format: 4Dの時間フォーマット。フォーマットが設定されていない場合、デフォルトのフォーマットが使用されます。値 = 4Dフォーマットの数字 (0 - 6). |
| <clipboard/> | クリップボードの内容で置換されるタグ。<br>*属性*:<br>- index: ペーストするクリップボード。値 = クリップボード番号 (0 - 9). |

* マクロはメソッドエディターのコンテキストメニューやタイプアヘッド機能を使用して呼び出すことができます (後述参照)。
** XML言語仕様に準拠するためには、拡張文字 (アクセント文字や引用符) を使用してはなりません。

以下はマクロ定義の例題です:

| マクロの内容 | コメント |
|---|---|
| <?xml version="1.0"...?> | XML宣言 |
| <!DOCTYPE macros SYSTEM> | ドキュメント宣言 |
| <macros> | マクロXMLファイルの開始 |
| <macro name="RecordLoop"> | マクロ定義と名前の開始 |
| <text> | マクロコードの開始 |
| For($i;1;Records in selection(<Selection/>)) | <Selection/>タグは、マクロ挿入時に4Dメソッド中で選択されたコードにより置換されます (例えばテーブル名) |
| SAVE RECORD(<Selection/>) | |
| NEXT RECORD(<Selection/>) | |
| End for | |
| </text> | マクロコードの終了 |
| </macro> | マクロ定義の終了 |
| </macros> | マクロXMLファイルの終了 |

### <method>タグについて

<method>タグを使用すると、4Dプロジェクトメソッドを実行するマクロコマンドを作成して使用することができます。これにより開発者はコンポーネントにマクロコマンドを添付して配布し、洗練された機能を提供することが可能になります。例えば以下のマクロはカレントのメソッド名を引数にとり、*MyMethod* メソッドを実行します:

```
<method>MyMethod("<method_name/>")</method>
```

呼び出されたメソッドコードは新規プロセスで実行されます。このプロセスはメソッド実行後に消失します。

**注:** 呼び出されたメソッドの実行が終了するまでストラクチャープロセスは停止されます。メソッドの実行は素早く終了し、アプリケーションをブロックするリスクがないことを確認しなければなりません。ブロックが発生してしまった場合、**Ctrl+F8** (Windows) または **Command+F8** (Mac OS) でこのプロセスをアボートできます。

### マクロを呼び出す

デフォルトでマクロはコンテキストメニュー、メソッドエディターのツールバー、自動補完機能、またはメソッドエディターウィンドウ下部のマクロリストから呼び出すことができます。

各マクロごとにコンテキストメニューや自動補完機能を使用した呼び出しを禁止することができる点に留意してください。

#### コンテキストメニューとツールバー

デフォルトですべてのマクロはメソッドエディターのコンテキストメニュー (**マクロ挿入**階層コマンド) またはツールバーのマクロボタンから呼び出すことができます。

<macro>タグの*in_menu*属性を使用して、マクロをこのメニューに表示させるかどうかを指定できます。

コンテキストメニュー内で、マクロは"Macros.xml"ファイルに記述された順番で表示されます。つまりファイルを編集することでこの順番を変更できます。

#### 自動補完

デフォルトですべてのマクロは自動補完 (タイプアヘッド) 機能からアクセスできます (メソッドを記述する参照)。<macro>タグの*type_ahead*属性を使用して、特定のマクロをこの動作の対象から外すことができます。

**注:** マクロに<selection/>タグが含まれている場合、このマクロは自動補完ポップアップウィンドウには表示されません。

#### メソッドエディターリスト

メソッドエディターのリストにマクロを表示させることができます (メソッドを記述する参照)。リスト中でマクロをダブルクリックし、呼び出すことができます。特定のマクロをこのリストから呼び出すことができます。

### 互換性に関する注意

マクロのサポートは4Dのバージョンごとに変わることがあります。カスタマイズされたマクロを管理しつつ、異なるバージョンの互換性を保持するために、4Dは以前のバージョンのマクロを取り除くことはしません。利用可能な最新の機能を使用したい場合は、それをあなたのバージョンに適用する必要があります。

#### methodの定義済みテキスト変数

v11より前のバージョンの4Dでは、<method>タグを使用する際、メソッド中でテキストを操作するための一連のプロセス変数が自動で管理されていました: 入力変数 (*_textSel*, *_blobSel*, *_selLen*, *_textMethod*, *_blobMethod*, *_methodLen*) を使用してテキストを取り出し、出力変数 (*_textReplace*, *_blobReplace*, *_action*) を使用してテキストを挿入しました。互換性のためこのメカニズムはまだサポートされています。しかしバージョン11からは以下の理由で廃止予定リストに置かれています:

- 32,000文字以上のテキストを扱うためにBLOB変数を使用する必要性がなくなった。
- 変数管理の方法が、変数の実行空間が区分けされるv11のアーキテクチャーと互換性がない。バージョン11のコンポーネントやホストデータベースは互いのメソッドの変数にアクセスできません。

ゆえに、選択されたテキストはGET MACRO PARAMETERとSET MACRO PARAMETERコマンドを使用して管理することが推奨されます。これらのコマンドを使用すればホストデータベースとコンポーネント間でテキストを相互に交換することができ、マクロを管理するためのコンポーネントを作成することができます。マクロでこのモードを有効にするには、macro要素内でversion属性に値"2"を指定しなければなりません。この場合、4Dは _textSel, _textReplace等の定義済み変数を管理しなくなり、GET MACRO PARAMETERとSET MACRO PARAMETERが利用可能になります。属性は以下のように定義します:

```
<macro name="MyMacro" version="2">
--- マクロテキスト ---
</macro>
```

この属性を指定しない場合、以前のモードが保持されます。

#### XML標準に関連する非互換性

4D v11より、マクロファイルはXML標準に準拠しなければならず、厳密なシンタックスルールが適用されるようになりました。これにより以前のバージョンで作成されたマクロコードに非互換が発生することがあり、またXMLファイルがロードできなくなることがあります。異常な動作が起こるのは以下のケースです:

- 以前のバージョンの4Dでは<macro>要素内で"// my comment"タイプのコメントが許可されていました。これはXMLのシンタックスに適合していません。このような行は標準の"<!-- my comment -->"形式で書きなおさなければなりません。
- インタープロセス変数名を表す<>記号はエンコードされなければなりません。例えば *<>params* 変数は<>paramsと書かなければなりません。
- 以前のバージョンの4Dではルート要素である<macros>を省略できました。現在このタグは必須です。このタグがない場合、ファイルをロードできません。

#### macros.xmlファイルの更新

4Dバージョン12では、SQLコマンドの利用を容易にするための新しいマクロコマンドを利用できます。"Macros.xml"ファイルは開発者がカスタマイズ可能なため、新しいバージョンの4Dをインストールしても自動では既存のファイルを更新しません。4D v12の新しいSQLマクロコマンドを使用するためには以下のいずれかの作業を行います:

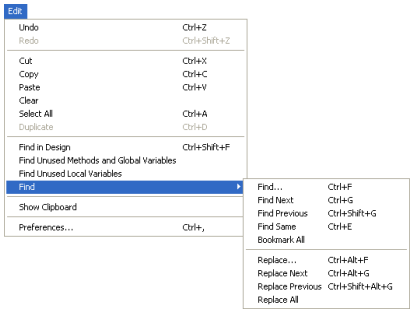
- "Macros v2"フォルダー内の"Macros.xml"ファイルを削除 (ファイルをカスタマイズしていない場合) して、4Dを起動し新しいファイルを作成させる。
- "Macros v2"フォルダー内の"Macros.xml"に手作業でで新しいマクロを追加する (ファイルをカスタマイズしている場合)。新しいテンプレートファイルは4Dアプリケーションフォルダーの4D\Resources\ja.lprojにあります。

## メソッド中の検索と置換

メソッドエディターには、カレントウィンドウに適用される、特別な検索/置換機能があります。

各ウィンドウのツールバーにある検索エリアを使用して単純な検索を実行したり、検索ダイアログを開いたりすることができます (メソッドエディター参照)。

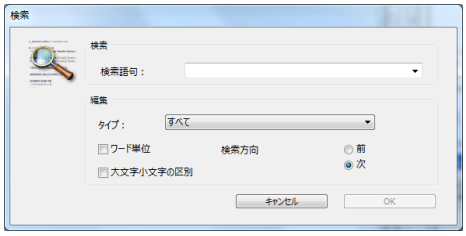
メソッド用の検索/置換コマンドは編集メニューの検索サブメニュー内にあります:




**注:** 検索サブメニューの上にある検索メニューはメソッドエディター固有のものではなく、すべてのメソッドを検索対象とするものです。詳細はデザインモードの検索と置換を参照してください。

### 検索

**検索...**コマンドを選択すると以下のダイアログボックスが表示されます:




このダイアログボックス内で定義された検索は最前面に位置するメソッドに対して適用されます。

- **検索語句**エリアには検索したい文字列を入力します。このエリアはコンボボックスで、最新の検索/置換語句を15個、4Dが終了するまで保持します。検索...コマンドを選択する前にエディター中でテキストをハイライトすると、このエリアにデフォルトでそのテキストが表示されます。
- **ワード単位**を選択すると、検索は (部分一致ではなく) 完全一致で行われます。例えば"client"を検索する場合、"clients"や"myclient"は検索されません。このオプションはデフォルトで選択されておらず、"var"を検索すると"Myvar"、"variation"等が検索されます。
  デザインモードを検索ダイアログボックスの**語全体を含む**オプションと異なり、**ワード単位**オプションはオブジェクト名を考慮に入れない点に留意してください。例えばこのオプションを選択した場合、"My"をメソッド中で検索すると"My Variable"変数が検索されます。**語全体を含む**オプションを使用してデータベース全体を検索した場合、上記のような結果にはなりません。なぜならオブジェクト名は"My Variable"であり、"My"とは一致しないからです。
- **大文字と小文字を区別する**オプションを使用すると、検索語句エリアに入力された文字の大文字と小文字が区別されます。例えば"MyVar"を検索した場合"myVar"はヒットしません。
- **前/次**ラジオボタンを使用して、現在カーソルのある方向を起点として、検索を行う方向を指定できます。

**OK**をクリックすると4Dは現在のカーソル位置から始めてメソッドの終端まで検索を実行します。そして最初に見つかった項目が選択されます。**編集**メニューの**次を検索/前を検索**コマンドを使用して、同じ条件の検索を繰り返すことができます。

#### 同じものを検索

**同じものを検索**コマンドを使用すると、選択した文字列と同じ文字列を検索することができます。このコマンドはメソッドエディター中で一文字以上選択した場合に利用可能となります。

実行される検索はカレントメソッド中の"次を検索"と同じです。

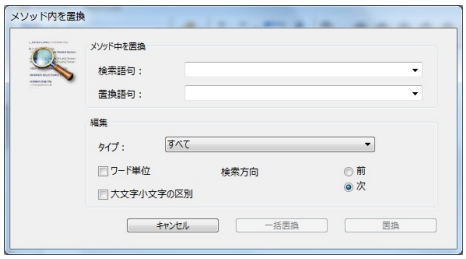
#### すべてをブックマーク

**すべてをブックマーク**コマンドは検索または置換ダイアログボックスで検索が定義されている場合に利用可能となります。このコマンドを選択すると、4Dは現在設定されている検索条件に当てはまる行にブックマークを設定します。これによりすべての検索結果に簡単に印をつけられます。

ブックマークに関する詳細はメソッドを記述するのブックマークを使用するを参照してください。

### 置換

**置換**コマンドを選択すると以下のダイアログボックスが表示されます:




- **検索語句**エリアには検索したい文字列や式を入力します。検索ダイアログと同様このエリアはコンボボックスで、最新の検索/置換語句を15個、4Dが終了するまで保持します。置換...コマンドを選択する前にエディター中でテキストをハイライトすると、このエリアにデフォルトでそのテキストが表示されます。
  **置換語句**エリアには検索語句エリアに入力された文字を置換する文字列を入力します。このエリアもコンボボックスで、最新の検索/置換語句を15 個、4Dが終了するまで保持します。
- **ワード単位**を選択すると、検索は (部分一致ではなく) 完全一致で行われます。例えば"client"を検索する場合、"clients"や"myclient"は検索されません。このオプションはデフォルト で選択されておらず、"var"を検索すると"Myvar"、"variation"等が検索されます。
- **大文字と小文字を区別する**オプションを使用すると、検索語句エリアに入力された文字の大文字と小文字が区別されます。例えば"MyVar"を検索した場合"myVar"はヒットしません。
- **前/次**ラジオボタンを使用して、現在カーソルのある方向を起点として、検索を行う方向を指定できます。

**置換**ボタンをクリックすると最初に見つかったオカレンスに対する置換が実行されます。4Dは現在のカーソル位置から始めて、メソッドの終端まで検索を行います。**編集**メニューの**次を置換/前を置換**コマンドを使用して、同じ条件の置換を繰り返すことができます。

**一括置換**ボタンをクリックすると開かれたメソッド中で検索条件に合致するすべてのオカレンスが一度で置換されます。

### 定義を開く

**定義を開く**コマンドはメソッドエディター中で参照されているオブジェクト (メソッド、フィールド、テーブル、フォーム) の定義を新しいウィンドウで開きます。これを行うにはカーソルをオブジェクト名の中に置き (または選択して)、**メソッド**メニューから**定義を開く...**を選択するか、エディターのコンテキストメニューを使用します。

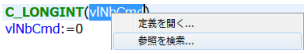
このコマンドは以下のオブジェクトに使用できます:

- **プロジェクトメソッド**: 新しいメソッドエディターを開き、メソッドを表示します。
  **注:** この機能は**Ctrl+K** (Windows) または **Command+K** (Mac OS) のキーボードショートカットでも利用できます。
- **フィールド**: ストラクチャーウィンドウのインスペクターにフィールドプロパティを表示します。
- **テーブル**: ストラクチャーウィンドウのインスペクターにテーブルプロパティを表示します。
- **フォーム**: フォームエディターでフォームを開きます。
- **変数** (ローカル、プロセス、インタープロセス、または$n引数): カレントメソッド中またはコンパイラーメソッド中の変数定義行を表示します。

**参照を検索**

**参照を検索...** コマンドは**メソッド**メニューまたはメソッドエディターのコンテキストメニュー内にあり、データベース中で、メソッドの現在の項目が参照されているすべてのオブジェクト (メソッドやフォーム) を検索します。

現在の項目とは選択されているもの、あるいはカーソルが存在するものです。フィールド名、変数名、コマンド、文字列等が該当します。例えば以下のアクションを実行するとデータベース中で*vlNbCmd*変数のオカレンスを探します:




このコマンドの結果は結果ウィンドウに表示されます。結果ウィンドウについては結果ウィンドウを参照してください。

**呼び出し元を検索**

**メソッド**メニューの**呼び出し元絵を検索**コマンドはプロジェクトメソッドでのみ利用可能です。このコマンドは、このプロジェクトメソッドを参照しているすべてのオブジェクト (他のメソッドやメニュー) を検索します。

**注: 呼び出し元絵を検索...** コマンドはエクスプローラーのメソッドページからも利用できます。

このコマンドは検索結果を新しい結果ウィンドウに表示します。結果ウィンドウに関する詳細は結果ウィンドウを参照してください。

**指定行に移動...**

この特別な検索コマンドは**メソッド**メニューの中にあります。このコマンドを選択すると、移動したい行番号を指定するためのダイアログボックスが開かれます。**OK**ボタンをクリックすると、コマンドはメソッド中で該当行を検索し、ハイライトします。このような検索はコンパイルデータベースで発生した実行時エラーで記録されるエラー行を表示するのに便利です。

メソッドエディターウィンドウに行番号を表示するかしないかを選択できます。このオプションについてはメソッドエディターを参照してください。

## メソッドの実行

アプリケーションに定義されたプロジェクトメソッドは通常ボタンやメニュー、あるいは他のメソッドから呼び出されます。データベースメソッドはアプリケーションで発生した特定のイベントに反応して実行されます。

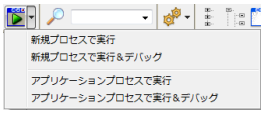
テストやデバッグの目的で、デザインモードでプロジェクトメソッドや特定のデータベースメソッドを実行することができます。この場合メソッドを新規プロセスで実行するか、または直接デバッグモードで一行ごと実行できます。

4D Server環境の場合、4D Serverマシン上でメソッドを実行するか、クライアントマシン上で実行するか指定できます。メソッド実行は2つの方法で行えます:

- メソッドエディターウィンドウから
- メソッド実行ダイアログボックスから (プロジェクトメソッドのみ)

**メソッドエディターからメソッド実行**

メソッドエディターウィンドウには、そのエディターで開かれているメソッドを実行するためのボタンがあります。このボタンに関連付けられているメニューから実行オプションを選択できます:




このボタンはプロジェクトメソッドと以下のデータベースメソッドでのみ利用できます:

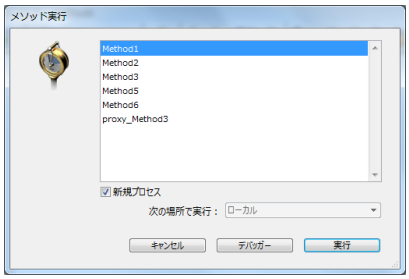
- On Startup
- On Exit
- On Server Startup
- On Server Shutdown

以下の実行モードから選択できます:

- **新規プロセスで実行**: プロセスを作成し、そのプロセス内でメソッドを実行します。
- **新規プロセスで実行＆デバッグ**: プロセスを作成し、デバッガーを開いてメソッドを表示します。
- **アプリケーションプロセスで実行**: アプリケーションプロセス内でメソッドを実行します。
- **アプリケーションプロセスで実行＆デバッグ**: アプリケーションプロセス内でデバッガーを開き、メソッドを表示します。

**メソッド実行ダイアログボックスからメソッド実行**

**実行**メニューから**メソッド...**コマンドを選択すると、メソッド実行ダイアログボックスが表示されます:




このダイアログボックスには非表示属性が設定されていないプロジェクトメソッド (コンポーネントの共有メソッドを含む) が表示されます。

プロジェクトメソッドを実行するには、リストからメソッドを選択し、**実行**をクリックします。デバッグモードでメソッドを実行するには**デバッグ**をクリックします。デバッガーに関する詳細は4Dランゲージリファレンスマニュアルのデバッグを参照してください。

**4D Serverに関する注記**: サーバー上でメソッドを実行する場合**デバッグ**オプションは使用できません。

**新規プロセス**チェックボックスを選択すると、選択したメソッドは新規に作成されたプロセス内で実行されます。大量の印刷など時間のかかる処理をメソッドが行う場合でもこのオプションを使用すれば、レコードの追加、グラフの作成などの処理をアプリケーションプロセスで継続できます。プロセスに関する詳細は4Dランゲージリファレンスマニュアルのプロセスを参照してください。

**4D Serverに関する注記:**

- クライアントマシンではなくサーバーマシン上でメソッドを実行したい場合、次の場所で実行メニューから**4D Server上**を選択します。この場合ストアドプロシージャーと呼ばれるプロセスが新規にサーバー上で作成され、メソッドが実行されます。
  このオプションを使用してネットワークトラフィックを減らしたり、4D Serverの動作を最適化したりできます (特にディスクに格納されたデータにアクセスする場合など)。すべてのタイプのメソッドをサーバー上や他のクライアント上で実行できますが、ユーザーインターフェースを変更するものは例外です。この場合ストアドプロシージャーは効果がありません。
- 他のクライアントマシン上でメソッドを実行するよう選択することもできます。他のクライアントマシンは事前に登録されていなければメニューに表示されません (詳細は4Dランゲージマニュアルの REGISTER CLIENTを参照)。
  デフォルトでは**ローカル**オプションが選択されています。4Dシングルユーザーの場合、このオプションしか選択できません。

# ユーザー＆グループ


アクセスシステムの概要

デザイナーと管理者

アクセスシステムを起動する

デフォルトユーザーを設定する

ユーザーとグループを管理する

ユーザにデザインモードへのアクセスを許可する

データベースオブジェクトにグループを割り当てる

パスワードアクセスシステムの保守

## アクセスシステムの概要

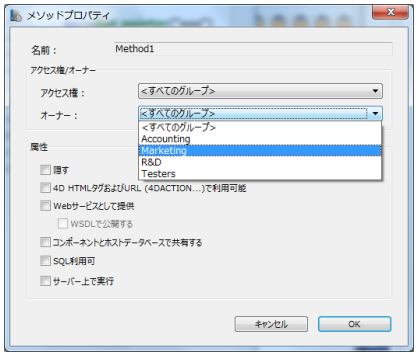
複数のユーザがデータベースを使用する場合は、データベースへのアクセスを制御したり、またはそれぞれのユーザに異なる機能やインタフェースを提供したいことがあります。マルチユーザ環境やWebで使用するアプリケーションを設計する場合、機密性の高いデータを保護することが重要になります。このためユーザにパスワードを割り当てたり、データベースデータやデータベース操作へのアクセスレベルが異なるアクセスグループを作成し、これらのデータを保護することができます。

### アクセス権

#### 権限を割り当てる

4Dのパスワードアクセスシステムは、ユーザーとグループに基づいています。ユーザーを作成してパスワードを割り当てたり、ユーザーをグループに入れて、各グループに対しデータベースの適切な部分へのアクセス権を割り当てます。グループにはアクセス可能なメソッドや、テーブル定義へのアクセス権が割り当てられます。

次の図は、あるプロジェクトメソッドにオーナーアクセス権を割り当てている様子を表わしています。グループにはアクセス権 (利用権限) やオーナー (更新権限) を割り当てることができます。



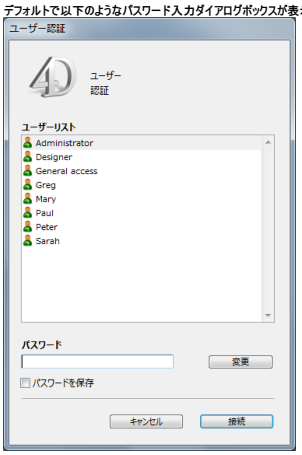

このポイントについてはパスワードアクセスシステムの保守で説明しています。

データベースを開くには、ユーザー名を選択するか入力し、パスワードを入力します。するとそのユーザーが属すグループと、そのグループに割り当てられたデータベースパーツに応じて、ユーザーはアクセスシステムで指定されたデータベースのパーツを操作できるようになります。

#### 保護されたデータベースへのアクセス

データベース設定のセキュリティページを使用してアクセス権を設定できます (セキュリティページ参照)。

- デフォルトで以下のようなパスワード入力ダイアログボックスが表示されます:

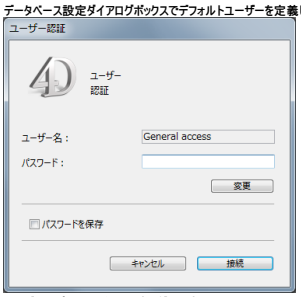



このダイアログボックスにおいて、ユーザーはユーザーリストから自分の名前を選択し、パスワード入力エリアに自分のパスワードを入力します。

- データベース設定ダイアログボックスで**パスワードダイアログにユーザーリストを表示する**が選択解除されている場合、以下のようなパスワード入力ダイアログが表示されます:

このダイアログボックスでは、ユーザーは自分の名前とパスワードの双方を入力しなければなりません。これにより、アプリケーションのセキュリティが強化されます。

- データベース設定ダイアログボックスでデフォルトユーザーを定義し、パスワードを割り当てた場合、次のダイアログボックスが表示されます:




ユーザーはパスワードを入力するだけです。

- デフォルトユーザーにパスワードを割り当てない場合、このダイアログボックスは表示されません。この場合各ユーザーにはデフォルトユーザーに定義されたものと同じ権限と制限事項が与えられます。

データベース設定で**ユーザーは自分のパスワードを変更可能**オプションが選択されている場合、パスワード入力ダイアログに**変更**ボタンが表示されます。このボタンをクリックして、リスト中で選択されているユーザーのパスワードを変更できます。

**4D Server に関する注記：** データベースへの接続識別情報を.4DLinkアクセスファイルに保存することができます。詳細は4D Serverリファレンスの4D Serverデータベースへの接続を参照してください。

ユーザーはデータベースを通常の方法で操作することができます。属しているグループが使用を許可されていないフォームやメニューコマンド、メソッド、テーブルをユーザーが使おうとすると、4Dは“お使いのパスワードではこのフォームを使用することができません”という旨のエラーメッセージを表示します。

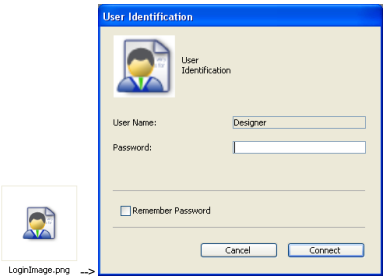
**注：** ON ERR CALLメソッドが設定されている場合、メソッドに関するエラーメッセージは表示されません。

#### IDウィンドウのアイコンのカスタマイズ

データベース接続ダイアログに表示されるアイコンをカスタマイズできます。デフォルトでは4Dロゴが表示されます。

このアイコンを変更するには、データベースの**Resources**フォルダー (データベースストラクチャーと同階層に配置、4Dファイルの説明参照)

にLoginImage.pngという名前で画像を保存します。形式はPNGでサイズは80x80ピクセルでなければなりません。




**パスワードを記憶する**

ユーザーが **パスワードを記憶する** オプションを選択した場合、4Dはダイアログボックスが閉じられたときにパスワードを保存し、次回以降の接続時にそれを再利用します。パスワードは暗号化され、データベースに対応する.4DLinkの中に保存されます。

このため、すべての4Dアプリケーションでこの機能が使用できるわけではありません。以下の表を参照してください:

| | 4Dローカルモード | 4Dリモートモード |
|---|---|---|
| エンジン組み込みでないデータベース | Yes | Yes |
| エンジン組み込みデータベース | No | Yes |

**4Dパスワードシステムを使用した外部からのアクセス**

ユーザーを定義したら、様々な外部アクセスから4Dデータベースへのアクセス制御に使用できます。特定のコンテキストではアクセス権の階層構造を利用することができます。

4Dのユーザー＆グループシステムを使用できるのは:

- **統合HTTPサーバー**: 詳細は接続セキュリティを参照してください。
- **統合SQLサーバー**: 詳細は4D SQLサーバの設定を参照してください。

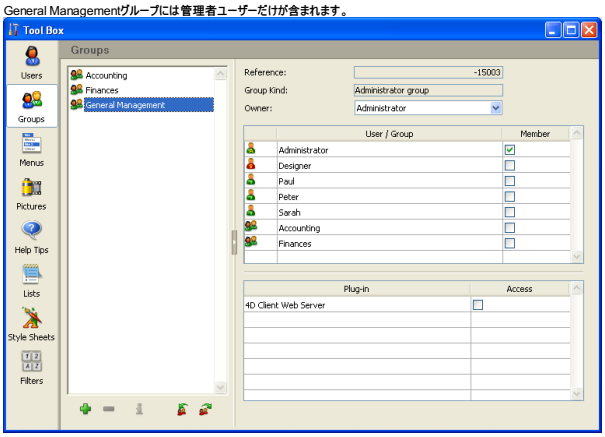
**アクセス権の階層構造**

データベースのセキュリティを確保し、ユーザーに異なるアクセスレベルを提供する最も効果的な方法は、アクセス権の階層構造を利用することです。ユーザーは適切なグループに割り振られ、各グループはネストされてアクセス権の階層構造を形成します。この節では、このような構造の取り扱い方について少し説明します。

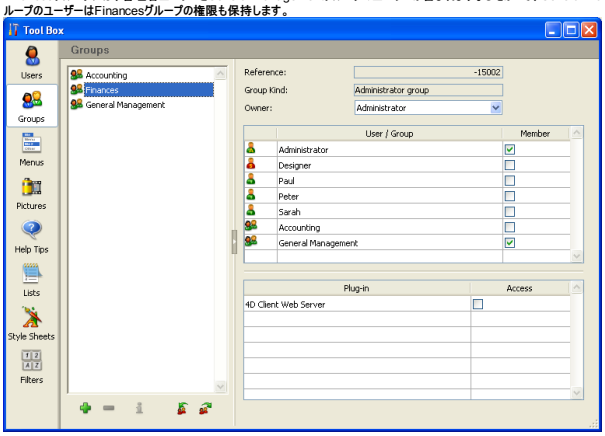
この例題では、ユーザーはそれぞれの担当にしたがい3 つのグループの1 つに割り振られます。会計グループに割り当てられたユーザーはデータ入力を担当します。財務グループに割り当てられたユーザーは、レコード更新や無効なデータの削除などのデータ管理を担当します。全般管理グループに割り当てられたユーザーは、検索の実行や分析レポートの印刷などのデータ分析を担当します。

グループが指定された時点で各権限が正しく配分されるように、各グループがネストされます。

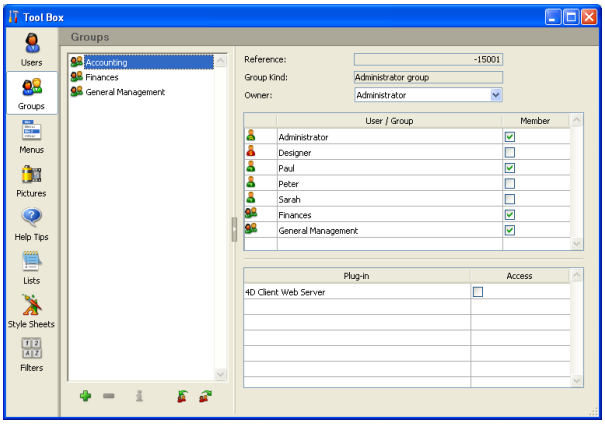
- General Managementグループには管理者ユーザーだけが含まれます。




- Financesグループには、管理者ユーザーとGeneral Managementグループのユーザーが含まれます。したがって、General Managementグループのユーザーは Financesグループの権限も保持します。




- Accounting グループには、データ入力を行うユーザーとFinances グループのユーザーが含まれます。したがって、Finances グループに属すユーザーとGeneral Managementグループのユーザーは Accounting グループの権限も利用できます。




それぞれの責務に基づいて、アクセス権を割り当てるグループを決定します。

例えばAccounting グループに入力フォームを割り当てると、すべての人がこの入力フォームを使用できるということになります。また、Financesグループに特定のフォームを割り当てた場合は、FinancesグループとGeneral Managementグループのメンバーだけがこのフォームを使用することができます。General Managementグループを割り当てると、このグループのメンバーしかそのフォームを使用することはできません。

このような階層システムを使用すると、新規ユーザーに割り当てるべきグループがわかりやすくなります。各ユーザーを1 つのグループに割り当てるだけで、グループの階層を使用してアクセス権を決定できます。

アクセススキームでは可能な限り低レベルでアクセスを制限すべきです。

## デザイナーと管理者

4Dは、ユーザーに対して2 つのモードへの標準的なアクセス権と、各モードにおける特定の権限を与えます。パスワードアクセスシステムを起動すると、これらの標準的な権限が有効になります。

最も強力なユーザーはデザイナー（Designer）です。デザイナーは、データベース設計を管理することができます。デザイナーはユーザーやグループを作成し、各グループにアクセス権を割り当て、デザインモードを使用することができます。デザイナーはデータベースに関するあらゆる操作を行うことができます。

デザイナーの次に強力なユーザーは管理者（Administrator）であり、通常はパスワードアクセスシステムを管理する役割を与えられています。初めてツールボックスの「ユーザー」ページを開くと、デザイナーと管理者がユーザーリストに表示されます。この時点では、特に「デザイン」モードへのアクセスが制限されていれば、管理者は特別なアクセス権を持たない通常ユーザーにすぎません。パスワードアクセスシステムを使用できるように、管理者には特別なアクセス権を与えなければなりません。この件に関する詳細は、パスワードアクセスシステムの保守の"メンテナンスのためのアクセス"を参照してください。管理者は、グループの保存や読み込みを行える唯一のユーザーです。管理者としてグループの保存や読み込みを行う方法についてはグループの読み込みと保存を参照してください。

他の部分への管理者のアクセスはグループのメンバーシップによって制限されます。これは、管理者がデータベース内でのアクセス権を得るためには、1 つ以上のグループに属さなければならないことを意味します。管理者はすべての新規グループに含まれますが、任意のグループから管理者の名前を取り除くことができます。

ユーザー管理のダイアログボックスにおいて、デザイナーと管理者のアイコンは、それぞれ赤色と緑色で表示されます:

- デザイナーアイコン:
- 管理者アイコン:

デザイナーと管理者の名前を変更することができますが、それぞれのアイコンを変えることはできません。

デザイナーや管理者により作成されたユーザーとグループは、それぞれのアイコンの色で識別することができます:

- デザイナーにより作成されたグループのアイコンは赤色であり、管理者により作成されたグループのアイコンは緑色です。
- デザイナーにより作成されたユーザーのアイコンの色は青であり、管理者により作成されたユーザーのアイコンの色は緑です。

グループオーナーはいつでもデフォルトの名前を変更することができます。

デザイナーと管理者は、それぞれ16,000 のグループと16,000 のユーザーを作成することができます。

## アクセスシステムを起動する

4Dのパスワードアクセスシステムは、デザイナーにパスワードを割り当てると起動します。

デザイナーにパスワードが指定されるまで、ユーザー＆グループを作成したとしても、すべてのユーザーが4Dデータベースのあらゆる部分を使用できます(データベースを開く際にIDを求められません)。

デザイナーにパスワードが指定されると、フォーム、メニュー、メソッドに割り当てられたすべてのアクセス権が有効になります。ユーザーがデータベースを開くには、パスワードを入力しなければなりません。

**警告：** デザイナーのパスワードは忘れないでください！これを忘れてしまうと、「デザイン」モードでデータベースを開けなくなります。

アクセスシステムを無効にするには、デザイナーのパスワードを削除します。

## デフォルトユーザーを設定する

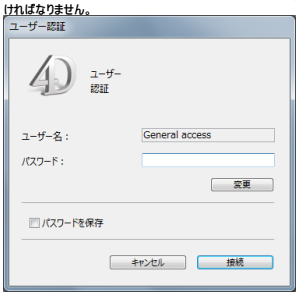
デフォルトユーザーを定義して、データベースを使用することができます。このオプションを選択すると、ユーザーがデータベースを開いたり、接続する際に名前を入力する必要がなくなります。さらに、デフォルトユーザーにパスワードを割り当てない場合は、パスワードダイアログボックスが表示されずにデータベースが直ちに開かれます。

デフォルトユーザーとしてログインした各ユーザーは、デフォルトユーザーに定義されたアクセス権と制約を保持します。このオプションにより、ユーザーアクションに関する管理システムはすべて維持したまま、データベースへ簡単にアクセスできるようになります。

デフォルトユーザーを定義するには、データベース設定のセキュリティページで、デフォルトユーザードロップダウンリストからユーザーを選択します。ツールボックスのユーザー＆グループエディターでユーザーを作成し、アクセス権や制約を設定しておく必要があります。

データベースへのアクセス権は、これ以上カスタマイズされません。データベースを開くと:

- デフォルトユーザーにパスワードを割り当てない場合、ログインダイアログボックスは表示されません。
- デフォルトユーザーにパスワードを割り当てると、データベースを開く際にダイアログボックスが表示され、デフォルトユーザーのパスワードを入力しなければなりません。




デフォルトユーザーモードが有効で、かつパスワードが必要となる場合は、環境設定ダイアログボックスのアクセスページで**ユーザーは自分のパスワードを変更可能**オプションの選択を解除しておくことをお勧めします。

### パスワードダイアログボックスを再表示する

4D に対し、標準のパスワードダイアログボックスを強制的に表示させることができます。例えば、デザイナーまたは管理者としてデータベースへ接続する場合です。

デフォルトユーザーモードが有効である場合に、パスワードダイアログボックスを再表示するには:

1. **Shift** キーを押しながらデータベースを開く。
   パスワードダイアログボックスが表示され、名前とパスワードを入力することができます。

## ユーザーとグループを管理する

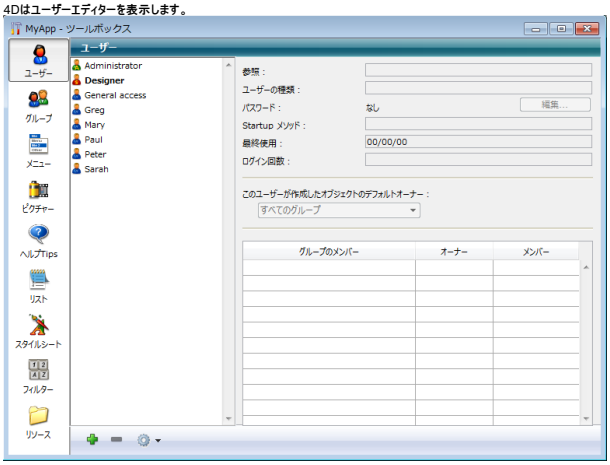
ユーザーとグループのエディターは4Dのツールボックスにあります。これらのエディターを使用して、ユーザーとグループを作成する、ユーザーパスワードを割り当てる、ユーザーをグループに入れるなどの操作を行うことができます。

### ユーザーの追加と変更

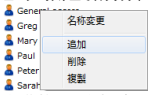
ユーザーエディターを使用して、ユーザーアカウントの作成やプロパティの設定、各グループへの割り当て、およびデータベースの使用状況の監視などの操作を行います。

ユーザーを追加するには:

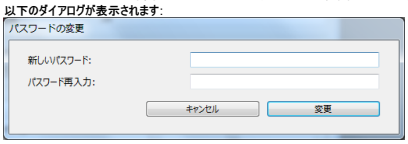
1. **デザイン**メニューから**ツールボックス＞ユーザー**を選択、または4Dツールバーの"ツールボックス"ボタンをクリックします。
   4Dはユーザーエディターを表示します。




   ユーザーリストには、カレントユーザーから"見える"すべてのユーザーが表示されます:
   - デザイナーの場合：すべてのユーザー
   - 管理者の場合：管理者が作成したユーザー（緑のアイコン）
2. ユーザーリストの下にある追加ボタンをクリックする。
   または
   ユーザーリスト上で右クリック（Windows）するか、またはControl+ クリック（Mac OS）し、コンテキストメニューから**追加**または**複製**を選択する。




   **注：** 複製コマンドを使用すると、同じ特性を持つ複数のユーザーを素早く作成することができます。
   4D は新規ユーザーをリストに追加し、デフォルトとして"新規ユーザーX"という名前を設定します。
   プロパティエリアには、このユーザータイプに関する情報が表示されます:
   - 参照エリアには選択したユーザーの参照番号が示されます。この番号はランゲージコマンドで使用します。
   - ユーザーの種類エリアにはそのユーザーのプロフィールの参照元が示されます。次のようなユーザータイプがあります:
     - デザイナー（Designer）：デザイナーユーザー
     - 管理者（Administrator）：管理者ユーザー
     - 開発者（Developer）：デザイナーが作成したユーザー
     - ユーザー（User）：管理者が作成したユーザー
3. 新しいユーザー名を入力する。
   この名前は、ユーザーがデータベースを開く際に使用されます。ユーザー名をいつでも変更することができます。変更するにはコンテキストメニューの**名称変更**コマンドを使用するか、**Alt+クリック**（Windows）または**Option+ クリック**（Mac OS）ショートカットを使用、または変更したい名前を2 回クリックします。
4. ユーザーのプロパティエリアで**編集...**ボタンをクリックして、パスワードの変更ダイアログを表示します。
   以下のダイアログが表示されます:




5. 新しいパスワード入力エリアにパスワードを入力し、確認エリアにそのパスワードをもう一度入力する。
   パスワードには15 桁までの英数字を使用することができます。パスワードのエディターでは大文字と小文字が区別されるため、ユーザーはここで入力した通りにパスワードを入力しなければなりません。例えば、あるユーザーのパスワードを"HolyCow"と定義した場合は、ユーザーはHとC を大文字にしてこのパスワードを入力しなければなりません。これを行わないと、4D はそのパスワードを受け入れません。
   パスワードを入力しても、ダイアログボックスには表示されません。入力した各文字の代わりにアスタリスクが表示されます。
6. ダイアログボックスを確定する。
   2 つのパスワードが一致しない場合、4D はビープ音を鳴らしてパスワードの変更をキャンセルします。
7. このユーザーが作成したオブジェクトのデフォルトオーナードロップダウンリストからグループを選択する。
   このグループはユーザーが作成したあらゆるオブジェクトを所有します（テーブル、フォーム、メソッド等）。例えばAccountingグループは、そのグループの各ユーザーにより作成されたオブジェクトを所有するように指定することができます。別のグループのユーザーが、Accountingグループのメンバーが作成したフォームを修正しようとすると、メッセージが表示され、そのユーザーにはそのフォームを編集する権限がないことを知らせます。
8. そのユーザーがデータベースを開くと実行される関連メソッドの名前を入力する（任意）。
   例えば、このメソッドは、ユーザー参照を読み込むために使用されます。
9. グループのメンバー表を用いて、そのユーザーが属すグループを設定する。
   - メンバーカラムの該当するオプションをチェックして、選択したユーザーをグループに追加することができます。また、このオプションのチェックを外すと、そのユーザーをグループから除外することができます。
   - オーナーカラムは、選択したユーザーがグループオーナーであるかどうかを示します。このカラムを変更することはできません。
   **注：** グループページのグループを使用し、各ユーザーを異なるグループのメンバーに設定することもできます。

既存ユーザーの特性を変更するには、リストからそのユーザーを選択して変更を行います。ユーザーのパラメータに関する情報は、前述のステップ3 から9 を参照してください。

### ユーザーを削除する

ユーザーを削除するには、そのユーザーを選択してから削除ボタンをクリックするか、またはコンテキストメニューの**削除**コマンドを使用します。

ユーザーアカウントを物理的に削除することはできません。アカウントの削除を要求すると、4D は操作の確認を求め、ユーザーリスト上でそのアカウントが削除済みユーザーとして表示されることを知らせます。
このダイアログボックスを確定すると、アカウントは無効になり使用できなくなります。

削除済みユーザーは、リスト上でグレー表示されます。




ユーザー参照番号を一意に保つため、この処理が必要になります。

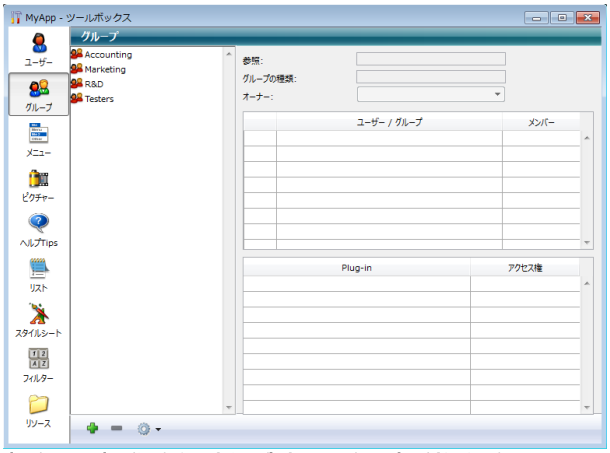
### アクセスグループを作成/設定する

グループエディターを使用して、各グループ内に納める要素（ユーザーや他のグループ）を設定したり、プラグインへのアクセス権を割り当てることができます。グループを作成する際、ユーザーのなかからグループのオーナーを指定することができます。

グループを作成してしまうと、削除できないということを覚えておいてください。グループを使用したくない場合は、そのグループに属すユーザーをすべて取り除きます。

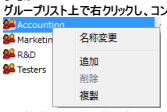
グループを作成するには:

1. **デザイン**メニューから**ツールボックス＞ユーザーグループ**を選択、または4Dツールバーのツールボックスボタンをクリックし、グループページを開く。
   4D はグループエディターウインドウを表示します:




グループリストには、グループを作成したユーザーを問わず、データベースのすべてのグループが表示されます。

2. グループリストの下にある追加ボタンをクリックする。
   または
   グループリスト上で右クリックし、コンテキストメニューから**追加**または**複製**を選択する。




   **注: 複製**コマンドを使用すると、同じ特性を持つ複数のグループを素早く作成することができます。
   4D は新規グループをリストに追加し、デフォルトとして"新規グループX"という名前を設定します。
   プロパティエリアには、このグループに関する情報が表示されます:
   - 参照エリアには選択したグループの参照番号が示されます。この番号はランゲージコマンドで使用します。
   - グループの種類エリアにはそのグループの参照元が示されます。次のようなグループタイプがあります:
     - デザイナーグループ: デザイナーが作成したグループ
     - 管理者グループ: 管理者が作成したグループ
3. 新しいグループの名前を入力する。
   グループ名には15 桁までの文字を使用できます。
   コンテキストメニューの名称変更コマンドを使用するか、**Alt+ クリック**(Windows)または**Option+ クリック**(Mac OS)ショートカットを使用、または変更したい名前を2 回クリックすると、グループ名をいつでも変更することができます。名前を変更できるのは、自分が作成したグループだけです。デザイナーは、管理者が作成したグループの名前を変更できません(この逆も同様)。
4. オーナードロップダウンリストからオーナーを選択する(任意)。
   グループオーナーは、そのグループのユーザーを追加したり、削除することができます(前述のグループオーナーの節を参照)。デフォルトでは、デザイナーが作成したグループであっても、管理者はすべてのグループのオーナーになるという点に留意してください。
5. メンバーエリアの該当するオプションをチェックして、グループのメンバーを設定する。
6. プラグインへのアクセス権を割り当てる(任意)。

最後の2 つのステップについては以下の節を参照してください。

#### グループオーナーを設定する

オーナーリストを使用して各グループのオーナーを指定することができます。通常オーナーは管理者ですが、グループメンバーならだれでもオーナーになることができます。

グループオーナーには、オーナーが所有するぐぅーぷからユーザーを追加したり取り除いたりする権限が与えられます。追加するユーザーは既存でなければなりません。グループオーナーはユーザーを作成することや、パスワードなどのユーザープロパティを変更することはできません。グループオーナーは他のグループを追加したり取り除いたりすることはできません。

デザインモードへのアクセスが制限されている場合、管理者のように、グループオーナーに対して明示的にパスワードエディターへのアクセス権を付与する必要があるかもしれません。この点に関する詳細はパスワードアクセスシステムの保守の"メンテナンスのためのアクセス"を参照してください。

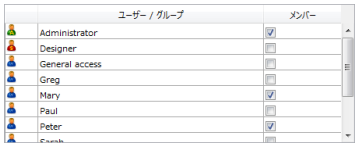
#### ユーザやグループをグループに入れる

任意のユーザーやグループをグループに配置することができます。さらに、そのグループ自体を他のいくつかのグループ内に入れることも可能です。ただし、必ずしもユーザーをグループに入れる必要はありません。

デザイナーはデータベース内のすべてのグループ内容を変更することができます。グループオーナーである管理者とユーザーは、自分がオーナーであるグループだけを変更できます。

しかし、ユーザーのステータスに関わらず、独自のグループから任意のユーザーやグループの表示、追加、削除を行うことができます。

ユーザーやグループをグループに配置するには、メンバー属性エリアの該当するオプションを選択します:




ユーザー名をチェックすると、そのユーザーがグループに追加されます。グループ名をチェックした場合は、そのグループの全ユーザーが新しいグループへ追加されます。
メンバーの一員となったユーザーやグループには、新しいグループに割り当てられたものと同じアクセス権が与えられます。

グループを別のグループ内に入れることにより、ユーザーの階層構造が作成されます。別グループに入れられたグループのユーザーは、両グループのアクセス権を保持します。例えば、総合管理グループを会計グループに入れると、総合管理グループの各ユーザーは、この両方の権限を取得します。しかし、会計グループだけに属しているユーザーは、そのグループの権限しか利用できません。アクセスシステム階層構造の操作に関する詳細はアクセスシステムの概要のアクセス権の階層構造を参照してください。

ユーザーやグループを別のグループから取り除くには、メンバー属性エリアの該当するオプションの選択を解除します。

#### プラグインや4D Client Web サーバにグループを割り当てる

データベースにインストールされたプラグインへのアクセス権をグループに割り当てることができます。これには4D のプラグインと任意のサードパーティプラグインが含まれます。プラグインに関する詳細はSet user properties を参照してください。

プラグインへのアクセス権を割り当てると、これらのプラグインに関して所有するライセンスの使用を管理できるようになります。プラグインのアクセスグループに属さないユーザーは、そのプラグインをロードすることができません。

また、プラグインアクセスエリアを使用して、4D Client Web サーバーとSOAPサーバーの使用を制限することも可能です。

ツールボックスのグループページにあるプラグインエリアには、4Dアプリケーションによりロードされたプラグインがすべて表示されます。プラグインへのアクセス権をグループに与えるには、該当するオプションをチェックします:

**4D Client Web Server**や**4D Client SOAP Server**項目を使用し、各4Dクライアントが Web 公開を行えるかどうかを管理することができます。4D Server 側では、クライアント側のWeb ライセンスがプラグインライセンスとしてみなされます。したがって、プラグインと同じ方法で、Web サーバライセンスの使用権を特定のユーザーグループに限定することができます。

### グループの読み込みと保存

4D では、管理者が自分で作成や変更を行ったグループを保存したり、読み込むことができます。グループを保存すると、カレントユーザーとカレントグループに関するあらゆる設定が保存されます。

管理者はグループを保存することができるため、データベースのアクセスシステムを保存して、そのシステムを同じデータベースの修正版や新規データベースに移行することができます。これは、新しいバージョンのデータベース用にアクセスシステムを再構築する上で非常に役立ちます。グループを再度読み込むことができるため、データベースのユーザーは新しいアクセスシステムを覚える必要がなくなります。

ユーザー名、パスワード、Startup メソッド名、グループ、グループオーナー、グループのメンバーはすべて保存されます。

**注:** デザイナーがグループエディターから保存や読み込みを行うことはできません。しかしデザイナーはUSERS TO BLOBやBLOB TO USERSコマンドを実行できます。

管理者が作成、変更したグループを保存するには:

1. 管理者としてデータベースを開始し、ツールボックスの**グループ**ページを表示する。
2. オプションメニューをクリックし、**ユーザー＆グループを保存**コマンドを選択する。
   4D はファイル作成用のダイアログボックスを表示し、ここでグループ名の設定とグループの保存を行うことができます。ユーザー＆グループファイルには".4UG"拡張子が付けられます。

グループを読み込むには:

1. 管理者としてデータベースを開始し、ツールボックスの**グループ**ページを表示する。
2. オプションメニューをクリックし、**ユーザー＆グループをロード**コマンドを選択する。
   4Dはユーザー＆グループファイルを選択するためのダイアログを表示するので、ロードするファイルを選択します。

## ユーザにデザインモードへのアクセスを許可する

すべてのユーザーはアプリケーションモードにアクセスできます。他方デザインモードへのアクセスを制限することができます。これを行うにはデータベース設定ダイアログボックスのセキュリティページにあるデザインアクセス権 ドロップダウンリストからグループを選択します。

その場合、このグループに属すユーザー、およびデザイナーだけがデータベースストラクチャを修正できるようになります。デザイナーは、明示的にストラクチャアクセスグループに属していなくても、常にデザインモードへアクセスすることができます。

その他のすべてのユーザーは、一般ユーザーです。ユーザーがデータベースを開くと、カスタムモードまたはユーザーモード（アクセスが認可されている場合）で開かれます。ユーザーのアクセス権は、属しているグループにより制限を受けます。

## データベースオブジェクトにグループを割り当てる

ユーザーとアクセスグループを定義したら、グループを次のオブジェクトに割り当てることができます:

- フォーム
- メソッド
- メニューコマンド

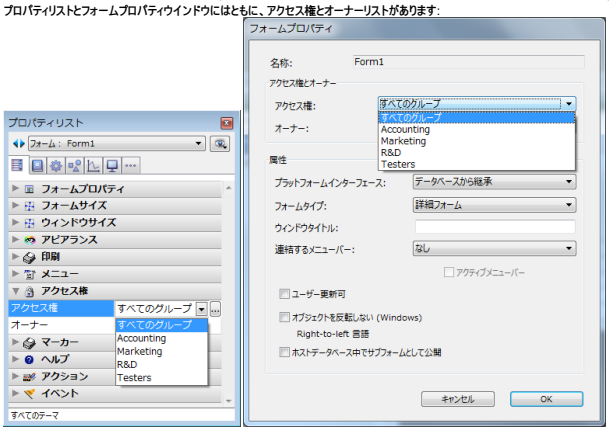
それぞれのオブジェクトに対し、グループは1 つしか割り当てることができません。このため、アクセス階層において、より強力なユーザーが自分より下の全グループに属すように、アクセスグループを設計することが重要になります。詳細はアクセスシステムの概要の"アクセス権構造スキーム"を参照してください。

### フォームにグループを割り当てる

フォームにグループのアクセス権を割り当てると、そのグループに属するユーザーだけがそのフォームを使用してデータを入力できるようになります。またフォームにグループのオーナー権を割り当てると、そのグループに属するユーザーだけがデザインモードでそのフォームを修正できるようになります。

アクセス権とオーナー権を割り当てるには:

1. フォームエディターでフォームを開き、プロパティリストにフォームのプロパティを表示する。
   または
   エクスプローラの**フォーム**ページでフォームを選択し、オプションメニューかコンテキストメニューから**フォームプロパティ...**コマンドを選択する。
   プロパティリストとフォームプロパティウインドウにはともに、アクセス権とオーナーリストがあります:




2. アクセス権ドロップダウンリストからグループを選択する。
   グループを選択しない場合は、すべてのグループがそのフォームへのアクセス権を保持します（デフォルト）。
3. オーナードロップダウンリストからグループを選択する。
   グループを選択しない場合は、すべてのグループがそのフォームへのオーナー権限を保持します（デフォルト）。

### プロジェクトメソッドにグループを割り当てる

プロジェクトメソッドにグループのアクセス権を割り当てると、そのグループに属するユーザだけがそのメソッドを使用できるようになります。プロジェクトメソッドにグループのオーナー権を割り当てた場合は、そのグループに属するユーザしかデザインモードでメソッドを修正できなくなります。

アクセス権とオーナー権を割り当てるには:

1. メソッドエディタでメソッドを開き、**メソッド**メニューから**メソッドプロパティ...**コマンドを選択する。
   または
   エクスプローラのメソッドページでプロジェクトメソッドを選択し、コンテキストメニューまたはオプションメニューから**メソッドプロパティ...**コマンドを選択する。
   メソッドプロパティダイアログボックスには、アクセス権とオーナーリストがあります (プロジェクトメソッドプロパティ参照)。
2. アクセス権ドロップダウンリストからグループを選択する。
   グループを選択しない場合は、すべてのグループがメソッドへのアクセス権を保持します（デフォルト）。
3. オーナードロップダウンリストからグループを選択する。
   グループを選択しない場合は、すべてのグループがメソッドへのオーナー権限を保持します（デフォルト）。

### メニューコマンドにグループを割り当てる

グループに属するユーザだけがアプリケーションモードでそのメニューコマンドを使用できるように、メニューコマンドにアクセスグループを割り当てることができます。

メニューコマンドにアクセスグループを割り当てるには:

1. **デザイン**メニューから**ツールボックス＞メニュー**を選択するか、4Dツールバーの"ツールボックス"ボタンをクリックし、**メニュー**ページを表示する。
   Tメニューバーエディタが表示されます。メニューバーエディターについてはメニューとメニューバーを参照してください。
2. メニューバーを選択する。
   このメニューバーに属するメニューが中央のリストに表示されます。
3. メニューバーリスト中のメニューを展開します。
   選択したメニューのメニューコマンドとメソッドが表示されます。
4. アクセスグループを設定したいメニューコマンドを選択する。
5. "アクセス権"ドロップダウンリストからグループを選択する。

## パスワードアクセスシステムの保守

パスワードアクセスシステムを有効にしたら、そのシステムを折に触れて保守する必要があります。ユーザーの追加や新しいグループメンバーの追加、パスワードの変更などの操作を行わなければなりません。デザイナーにはデザインモードへのアクセス権があり、ツールボックスのユーザ＆グループページを用いて、必要な変更を実行することができます。

また管理者とデザイナはー、保守の必要に応じて各ユーザーのデータベース利用履歴を調べることができます。

**注：** メンテナンスの目的やデータベースの更新を容易にするために、ユーザー＆グループを書き出して読み込むことができます。詳細はユーザーとグループを管理するを参照してください。

### 保守のためのアクセス

管理者がデザインモードへのアクセス権を持たない場合もあります (WA SET PREFERENCE参照)。しかし、デザイナーがEDIT ACCESSコマンドを含むプロジェクトメソッドを作成すると、管理者とグループオーナーは限られた範囲でユーザーやグループを制御できるようになります。

EDIT ACCESSコマンドは、カスタムメニューに割り当てるメソッドに記述することが可能であり、また**実行**メニューの**メソッド...**コマンドを選択して実行できます。管理者でも別のグループオーナーでもないユーザーがこのメソッドを実行しても、メソッドは何も行いません。

このメソッドが実行された場合、ユーザーが管理者またはグループオーナーのいずれであるかによって、その結果は異なります。

- 管理者がEDIT ACCESSコマンドを含むメソッドを実行すると、4D はユーザーとグループのエディターだけが含まれるツールボックスを表示します。管理者はこれらのエディターを使用してユーザーやグループを作成することができます。さらに、自分が作成したユーザーやグループを編集して、ユーザーパスワードを変更したり、作成したグループのユーザーを追加したり、削除することができます。管理者はフォームやメニューコマンド、メソッド、プラグインにグループを割り当てることはできません。デザイナーだけがこれらのアクセスグループを割り当てることができます。
- 管理者以外のグループオーナーがEDIT ACCESSコマンドを含むメソッドを実行すると、4D はグループエディターだけが含まれるツールボックスを表示します。さらにエディターにはそのユーザーがオーナーであるグループだけが表示されます。グループオーナーは、そのグループのユーザーを追加したり、削除することができますが、ユーザーの作成やユーザー情報の編集、またはグループの追加は行えません。ユーザーやグループの追加や編集を行うメニューコマンドはグレー表示されます。

### 利用状況を調べる

ツールボックスのユーザーページには、ユーザーがそのデータベースを前回使用した日付と利用（ログイン）回数が表示されます。管理者やデザイナーはリストからユーザーを選択して、この情報を調べることができます。

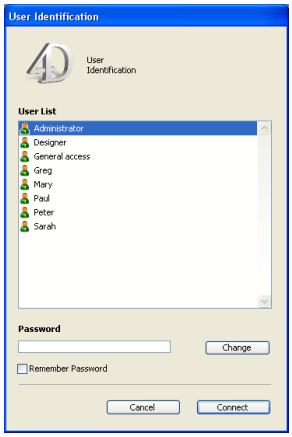
ユーザー情報を表示するには:

1. デザインメニューから**ツールボックス＞ユーザー**を選択するか4Dツールバーの"ツールボックスボタンをクリックする。
   または
   EDIT ACCESSコマンドを含むメソッドを実行する。
   4D はユーザーエディターを含むツールボックスを表示します。
2. 調べたいユーザーの名前をユーザーリストから選択する。
   ダイアログボックスには、そのユーザーがデータベースを前回使用した日付と、そのデータベースを開いた回数が表示されます。

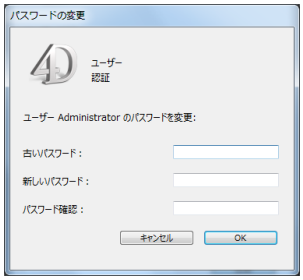
| | |
|---|---|
| 最終使用： | 2012年3月30日 |
| ログイン回数： | 27 |

### ユーザーによるパスワードの変更

ユーザー認証ダイアログボックスにはデフォルトで**変更**ボタンが表示され、カレントユーザーは自分のパスワードを変更できます。ユーザーリストが表示されている場合、選択されているユーザーがカレントユーザーです。リストが表示されていない場合、ユーザー認証ダイアログボックスにカレントユーザーの名前を入力しなければなりません:




ユーザーが変更ボタンをクリックすると、以下のダイアログボックスが表示されます:




パスワードを変更するにはまず現在のパスワードを"古いパスワード"欄に入力します。そして新しいパスワードを"新しいパスワード"と"パスワード確認"欄に入力します。ダイアログを確定すると、入力が正しければ、パスワードが変更されます。ユーザーは新しいパスワードを入力して**接続**ボタンをクリックし、データベースに接続します。

**変更**ボタンを非表示にして、ユーザーによるパスワードの変更を防止できます。このオプションはデータベース設定のセキュリティページにあります。

# メニューとメニューバー


メニューインターフェースを設計する

メニューエディター

メニューを構築する

メニューを添付する

メニュータイトルに参照を使用する

メニューアクションを指定する

メニュープロパティを設定する

メニューバーの管理

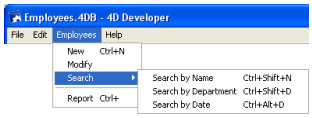
# メニューインターフェースを設計する

データベースやカスタムアプリケーション用にカスタムメニューを作成できます。デスクトップアプリケーションではプルダウン形式のメニューが標準機能であるため、メニューを追加することであなたのデータベースがより使いやすくなりユーザーに親しみやすいものになるでしょう。カスタムメニューを作成する場合、独自のツールバーを作成することもできます。独自のメニューとツールバーを使用すれば、お使いのデータベースは"スタンドアロン"アプリケーションのように機能します。

カスタムアプリケーションには、少なくとも1 つのメニューバーを作成し、そこに少なくとも1 つのメニューを納めなければなりません。新規データベースを作成すると、デフォルトとして4D はカスタムメニューバーを1 つ作成するため、これを使用してアプリケーションモードへアクセスできるようになります。カスタムアプリケーションの作成に関する詳細は4D ランゲージリファレンスマニュアルの4Dアプリケーションのビルドを参照してください。

## 概要

一般的に、メニューはユーザがデータベース処理を行うために選択するメニューコマンドを提供します。例えば、レコードの修正、レコードの検索、レポートの印刷等の処理です。次の図は、カスタムメニューの例を示しています:




メニューバーはスクリーン上にまとめて表示されるメニューグループです。メニューバー上の各メニューはメニューコマンドを持ちます。またメニューコマンドは階層メニューと呼ばれるサブメニューを持つこともできます。メニューやサブメニューコマンドをユーザーが選択すると、プロジェクトメソッドまたは標準アクションが呼び出されます。

各データベースに対し、異なるメニューバーを複数作成することもできます。例えば、ひとつのメニューバーを使用して標準的なデータベース処理用のメニューを納め、別のメニューバーはレポート作成時にのみアクティブにすることができます。また、あるメニューバーにはレコード入力用のメニューコマンドを含むメニューを格納することも可能です。入力フォームと一緒に表示されるメニューバーには同じメニューを格納しながらも、データ入力中は不要になるメニューコマンドを選択不可にすることができます。

さらに、メニューバーエディタを使用して、カスタムツールバーを作成することもできます。これを行うには、メニューコマンドにアイコンを割り当てます。このアイコンは4D のツールバー上に表示され、そのメニューコマンドのテキストがアイコンのヒントとして利用されます。

あるメニューを複数のメニューバーで使用したり、どのメニューバーにも割り当てずにプログラムからのみ管理することもできます (独立メニューと呼びます)。

メニューを設計する際には以下の2つのルールを覚えておいてください:

- メニューに適している機能に対しメニューを使用する : メニューコマンドは、レコードの追加や検索、レポートの印刷のような作業を実行しなければなりません。
- メニューコマンドを機能別にまとめる : 例えば、レポートの印刷を行うメニューコマンドはすべて同じメニュー内に置くべきです。また別の例として、特定のテーブルに関するすべての操作を1 つのメニューに納めてもよいでしょう。

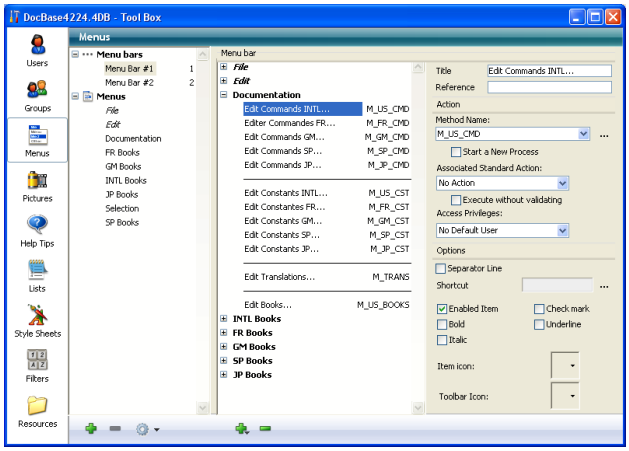
メニューやメニューバーを作成するには以下のいずれかを使用します:

- メニューエディター
- ランゲージコマンド
- 上2つの組み合わせ

メニューバーエディターを使用せず、4Dランゲージを使用してプログラムからメニューやメニューバーを作成し管理することもできます。この点に関する詳細は4Dランゲージリファレンスマニュアルのメニューを参照してください。

## メニューエディター

ツールボックスのメニューページを開くと、メニューエディターにアクセスできます。

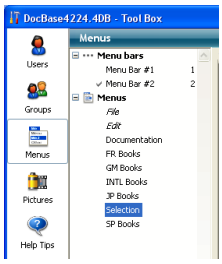



メニューやメニューバーは両方ともエディター左の階層リスト中に表示されます。これによりデータベース中に定義されたすべてのメニューを1つのリストで見ることができます。メニューバーを選択する必要はありません。メニューは文字順に表示されます。

各メニューはあるメニューバーあるいは他のメニューに添付できます。後者の場合、そのメニューはサブメニューとなります (メニューに他のメニューを添付する (サブメニュー)参照)。

メニューバーやメニューの内容を表示するには、エディター左のリスト中でそれをクリックします。そのメニューバーやメニューに属する項目が中央のエリアに表示されます。メニューバーやメニューのプロパティがウィンドウの右側に表示されます。メニュー項目のプロパティを表示するにはウィンドウ中央のリスト内でそれを選択します。

メニューを選択した場合、そのメニューが添付されているメニューバーやメニューにはチェックマークが表示されます。メニューが複数の項目に添付されている場合、複数のチェックマークが表示されます。以下の例では"Selection"メニューがMenu Bar #2に添付されています:




メニューが使用されていない場合 (独立メニュー)、チェックマークは表示されません。

### コンテキストメニューとオプションメニュー

メニューエディターにはコンテキストメニューがあり、クリックされた項目のタイプ (メニューバー、メニュー、メニュー項目) に応じて利用可能なアクションに直接アクセスできます。コンテキストメニューを使用して項目を追加したり削除したり、リストを展開したり折りたたんだりできます。

メニューエディターにはオプションメニューも用意されています。このメニューにはウィンドウ左下部の歯車ボタンをクリックしてアクセスします。このメニューにはメニューエディター標準コマンドとコンテキストコマンドが含まれます。標準コマンドを使用して新規メニューバーや新規メニューを作成したり、標準の編集メニューを作成したりできます。コンテキストメニューは選択した項目 (メニューバーまたはメニュー) に応じて変化し、対応するアクションを表示します。

# メニューを構築する

### デフォルトメニューバー

新規にデータベースを作成すると、4Dは自動でデフォルトメニューバー (メニューバー番号1) を作成します。このメニューバーには標準のメニューとデザインモードに入るためのコマンドが用意されています。

このメニューが用意されているため、ユーザーはデータベースを起動するとすぐにアプリケーションモードを使用できます。**実行**メニューから**アプリケーションテスト**コマンドを選択すると、自動でメニューバー番号1が呼び出されます。

デフォルトのメニューバーには**ファイル**、**編集**、そして**モード**メニューがあります。

- **ファイル**: このメニューには**終了**コマンドだけが含まれています。このコマンドには終了標準アクションが割り当てられていて、選択されるとアプリケーションが終了します。
- **編集** (標準): 編集メニューは標準であり、内容の変更が可能です。編集メニューのコマンド (コピーやペーストなど) は標準アクションで指定できます。
- **モード**: モードメニューにはデフォルトで、アプリケーションモードを終了するための**デザインモードに戻る**コマンドが含まれます。

**注:** メニュータイトルはハードコードされたテキストではなく、xliff参照を使用しています。この点についてはメニュータイトルに参照を使用するを参照してください。

このメニューを必要に応じて変更したり、新しく追加したりできます。

### メニューバーを作成する

ここではカスタムメニューバーを作成する方法について説明します。

**注:** プログラムだけでメニューバーを作成できます。詳細は4Dランゲージリファレンスマニュアルのメニューを参照してください。

メニューバーを作成するには:

1. 4Dツールボックスのメニューバーページを表示します。
   4Dはメニューバーエディターを表示します。作成済みのメニューバーが左のパネルに表示されます。
   4Dはメニューバーに連番を割り当てます。メニューバー番号1が一番上に表示されます。メニューバーの名前を変更することができますが、番号は変更できません。この番号はランゲージコマンドで使用されます。
2. メニューバーエリア下にある追加ボタンをクリックします。
   または
   リストのコンテキストメニューまたはリストの下にあるオプションメニューから**新規メニューバー作成**を選択します。
   新しいメニューバーがリスト中に追加されます。デフォルトメニュー (ファイルと編集) があらかじめ添付されています。
3. 必要であれば名前を変更します。
   メニューバーの名前上で**Alt+クリック** (Windows) または **Option+クリック** (Mac OS) するか、名前上で2回クリックすると、名前を編集できるモードになります。
   または
   ウィンドウ右の"タイトル"エリアに名前を入力します。
   適切な名前を付けると、デザインモードの別のエディターやランゲージコマンドでメニューを識別することが容易になります。メニューバー名は31文字まで指定でき、ユニークでなければなりません。
   この時点でメニューバーを変更したり、メニューやメニューコマンドを追加したりできます。

### メニューを作成する

いつでもメニューを作成できます。作成したメニューはメニューバーや他のメニュー (階層サブメニュー) に添付したり、あるいはプログラムから使用したりすることが可能です。

表示順にメニューを作成する必要はありません。作成後にドラッグ＆ドロップで並び替えることができます ("メニューやメニューコマンドの並び替え参照")。

**注:** メニューを完全にプログラムで作成することができます。この点に関する詳細は4Dランゲージリファレンスマニュアルのメニューを参照してください。

メニューを作成するには:

1. 4Dツールボックスの"メニュー"ページを表示します。
2. ウィンドウ左の階層メニュー中で"メニュー"またはその子項目の既存のメニューを選択し、ボタンをクリックします。
   または
   コンテキストメニューまたはエディターのオプションメニューから**新規メニュー作成**を選択します。
   4Dはリストの最後に新しいメニューを追加します。
3. メニュー名上で**Alt+クリック** (Windows) または **Option+クリック** (Mac OS) するか、2回クリックすると、名前編集モードに入ります。
   または
   "タイトル"エリアに入力します。
   テキストリテラル、変数参照、またはxliff参照を使用できます。この点についてはメニュータイトルに参照を使用するを参照してください。
   名前を直接入力する場合、制御文字が挿入されないように留意してください。メニューの表示が妨げられることがあります ("メニューラベルに制御文字を使用する"参照)。
4. メニューをさらに追加するにはステップ2と3を繰り返します。
   作成したメニューをプレビューするには、ソースメニューを選択し、エディターウィンドウ右側のプレビューエリアに表示されるメニューをクリックします。

### メニュー項目を作成する

各メニューには、メニューがクリックされたときにドロップダウンされるメニューコマンドを作成しなければなりません。項目を追加してメソッドや標準アクションを割り当てたり、他のメニューをサブメニューとして添付したりできます。階層メニューの添付はメニューを添付するで説明します。

メニューコマンドを追加するには:

1. ソースメニューリスト中で、コマンドを追加するメニューを選択します。
   メニューが既にコマンドを持っていれば、それが中央のリストに表示されます。新しいコマンドを挿入するには、コマンドを挿入したい位置のコマンドを選択します。後で順番を変更することも可能です。
2. メニューエディターのオプションメニューまたはエディターのコンテキストメニュー (中央のリスト内で右クリック) から"**メニュー "メニュー名" に項目を追加**"を選択します。
   または
   中央のリストの下にある追加ボタンをクリックします。
   項目が追加され、デフォルト名"項目 X"が割り当てられます (Xは項目の番号)。
3. コマンド名上で**Alt+click** (Windows) または **Option+click** (Mac OS) 、あるいは2回クリックすると名前編集モードに入ります。
   または
   "タイトル"エリアに名前を入力します。
   テキストリテラル、変数参照、またはxliff参照を使用できます。この点についてはメニュータイトルに参照を使用するを参照してください。
   名前を直接入力する場合、制御文字が挿入されないように留意してください。メニューの表示が妨げられることがあります ("メニューラベルに制御文字を使用する"参照)。
4. (オプション) "パラメーター"エリアにカスタム参照を入力します。
   この参照はランゲージコマンドから使用できます。詳細はカスタムパラメーターを参照してください。
5. さらにコマンドを追加するにはステップ1から4を繰り返します。

### メニュータイトルに制御文字を使用する

メニューコマンドのラベルに制御文字（メタ文字）を直接使用し、メニューコマンドのプロパティを定義することができます。例えば、メニューコマンドのラベルに"/G"という文字を入れると、キーボードショートカットであるCtrl+G（Windows）またはCommand+G（Mac OS）をメニューコマンドに割り当てることができます。

制御文字はメニューコマンドのラベルには表示されません。したがって、制御文字として使用しない場合は、これらの文字を使用していないことを確認する必要があります。制御文字には次のようなものがあります:

- ( (開く括弧)
- < (小なり)
- ! (感嘆符)
- ^ (キャレット)
- / (スラッシュ)

これらの文字の役割に関する詳細は4DランゲージリファレンスマニュアルのAPPEND MENU ITEMコマンドの説明を参照してください。

### メニューやメニューコマンドの並びを入れ替える

メニューバーのメニューやメニュー用のメニューコマンドを作成した後で、その順序をドロップ＆ドロップ操作を用いて変更することができます。メニューコマンドを順序の異なる場所に挿入するには、そのメニューコマンドを新しい位置にドラッグします。また、メニューを移動するには、そのメニューをメニューリスト上の別の位置にドラッグします。

### メニューやメニューコマンドを削除する

メニューバー、メニュー、メニューコマンドを削除できます。各メニューやメニューバーはひとつの参照しか持たない点に留意してください。メニューが複数のメニューバーやメニューに添付されていた場合 (メニューを添付する参照)、そのメニューに対して行われた変更や削除はこのメニューのすべての他のオカレンスに対しても有効となります。

メニューバー、メニュー、メニューコマンドを削除するには2つの方法があります:

- 削除する項目を選択し、リストの下にある削除ボタンをクリックします。
- こんてきすとめにゅーまたはエディターのオプションメニューから**メニューバー "メニューバー名" を削除**、**メニュー "メニュー名" を削除** または **項目 "項目名" を削除** コマンドを選択します。

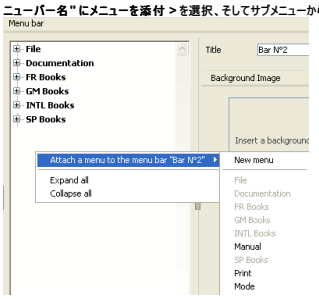
**注:** 最後のメニューバーを削除することはできません。

## メニューを添付する

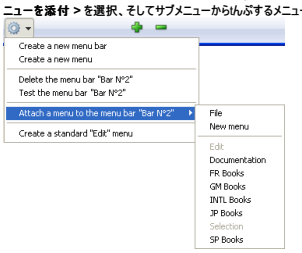
### メニューバーにメニューを添付する

メニューを作成したら、それをメニューバーに添付できます。これはドラッグ＆ドロップ、オプションメニュー、あるいは中央エリアのコンテキストメニューから行えます。

- **ドラッグ＆ドロップ**: メニューバーをクリックし、その内容を中央のリストに表示します。そして左のリストからメニューを選択して、中央リストにドラッグします。
- **コンテキストメニュー**:メニューバーをクリックし、その内容を中央のリストに表示します。中央リストの空白部分で右クリックし、**メニューバー "メニューバー名" にメニューを添付 >** を選択、そしてサブメニューからけんぷするメニューを選択します:




- **オプションメニュー**: 左のリストからメニューバーを選択し、リストの下にあるオプションメニューをクリックします。**メニューバー "メニューバー名" にメニューを添付 >** を選択、そしてサブメニューからけんぷするメニューを選択します:

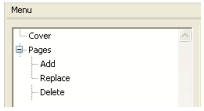



### メニューに他のメニューを添付する (サブメニュー)

メニューを他のメニューに添付し、階層メニューを構成できます。メニューバーでは、同じメニュー内で、テーマに基づき機能をグループ化する目的でサブメニューが使用されます:

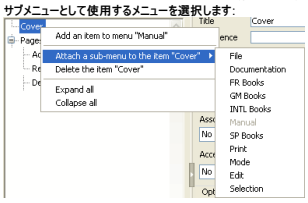
サブメニューとその項目はメニュー自身と同じ属性 (アクション、メソッド、ショートカット、アイコン等) を持つことができます。

メニューエディター中で、サブメニューは階層リスト項目として表示されます:




サブメニューを作成するには、既存のメニューを他のメニューに添付します。これを行うには2つの方法があります:

- **ドラッグ＆ドロップ**: 左側のリストからメニューを選択し、中央のリスト項目中サブメニューを添付したい項目にドラッグします。
- **コンテキストメニュー**: 中央のリスト中でメニューを添付したい項目上で右クリックし、**項目 "項目名" にサブメニューを添付 >** を選択、そしてサブメニューとして使用するメニューを選択します:




添付されたメニューはサブメニューとなります。項目のタイトルは保持されます (元のサブメニュー名は無視されます) がこのタイトルを変更することができます。

サブメニューの項目は元の特性やプロパティを保持し、サブメニューの動作は標準のメニューと同じです。

サブメニューのサブメニューを作成できます。中央のリスト中でサブメニューを展開し、メニューを添付します。理論的には階層の制限なくサブメニューを添付できます。しかしインターフェース標準に沿うためには、2レベルを超えるサブメニューは推奨されません。

### メニューやサブメニューを分離する

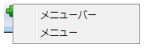
メニューバーからメニューを、あるいはメニューからサブメニューを分離できます。分離されたメニューはメニューバーやメニューから利用できなくなります。しかしメニューリストには残されます。

メニューを分離するには、中央リスト中メニューを分離したいメニュー上で右クリックし、**メニュー "メニュー名" をメニューバー "メニューバー名" から分離**または**項目 "項目名" のサブメニューを分離**を選択します。

### 独立メニュー

メニューバーや他のメニューバーに添付されていないメニューを独立メニューと呼びます。このようなメニューをメニューエディターで作成し、ランゲージコマンドで管理できます (メニュー参照)。

独立メニューを作成するには、メニューバー/メニューリストの下にある追加ボタンに割り当てられたメニューから、**メニュー**を選択します:



## メニュータイトルに参照を使用する

メニューのタイトルとして文字列リテラルではなく参照を使用できます。これによりアプリケーションの翻訳が容易になります。3タイプの参照を使用できます:

- **XLIFFリソース参照**: *:xliff:MyLabel* や *:15115,3* (互換性)。XLIFF参照については付録 C: XLIFFアーキテクチャーを参照してください。
- **インタープロセス変数名**: *:<>vlang,3*。この変数の内容を変更すると、メニューが表示される際、ラベルが変更されます。この場合データベースの翻訳アーキテクチャーに基づき、ラベルはXLIFFまたはSTR#リソースを呼び出します:
  - *XLIFFアーキテクチャー*: <>vlang変数に含まれる値はgroup要素のid属性値に対応します。2番目の値 (例では3) はtrans-unit要素のid値を指定します。
  - *STR#アーキテクチャー*: <>vlang変数はSTR#リソース番号、2番目の番号はそのリソース内でのリソース項目番号を表します。
- **STR#リソース参照**: 例えば":20000,3"と記述すると、STR# 2000リソースの3番目の項目がメニュータイトルとして表示されます。このSTR#リソースの内容を4Dランゲージを使用してプログラムで、あるいは (Mac OSで) リソースエディターを使用して更新すると、メニューが表示される際、ラベルが変更されます。

**互換性に関する注記:** Macリソースに基づく参照は互換性のために保持されています。このメカニズムは廃止予定であり、その利用は推奨されません。XLIFFアーキテクチャーの利用を推奨します。

## メニューアクションを指定する

メニューコマンドがその機能を実行するためには、プロジェクトメソッドや標準アクションを割り当てなければなりません。

メニューコマンドにより表わされる各機能は、これらのメソッドや標準アクションにより実行されます。例えば、"**月次報告書**"メニューコマンドは、財務データを格納したテーブルをもとに月次報告書を作成するプロジェクトメソッドを呼び出します。"**カット**"メニューコマンドはカット標準アクションを呼び出して、選択項目をクリップボードへ移動し、それを前面にあるウインドウから消去します。メニューコマンドが選択されると、4D は割り当てられたプロジェクトメソッドを実行します。

求める結果の種類によって、標準アクションまたはプロジェクトメソッドのいずれを割り当てるかを選択します。原則として、標準アクションは最適化された方法で実行される (特にコンテキストに応じてメニューの有効/無効が自動で切り替えられる) ため、できるだけこちらを選ぶ方が良いでしょう。

さらに、標準アクションとプロジェクトメソッドの双方をメニューコマンドに割り当てることも可能です。この場合、標準アクションが実行されることはありません。しかし、4D はこのアクションを使用し、状況に合わせてメニューコマンドを使用可／使用不可に設定します。メニューコマンドが使用不可の場合、割り当てられたプロジェクトメソッドは実行されません。

プロジェクトメソッドはメソッドエディターで作成します。これらのメソッドの作成は、メニューコマンドへの割り当て前または後のいずれでも行えます。メニューバーエディターでメニューコマンドにメソッドを割り当てたら、このメソッドを**[...]**ボタンで開くことができます。

メニューコマンドに標準アクションやメソッドが割り当てられていない場合、カスタムモードでそのメニューコマンドを選択すると、4D は自動的にユーザーモードへ戻ります (許可されていれば)。

### プロジェクトメソッドを割り当てる

メニューコマンドにプロジェクトメソッドを割り当てるには:

1. メニューコマンドを作成するか、または選択する。
   プロパティエリアが変更されて、選択したメニューコマンドのプロパティが表示されます。
2. データベースにプロジェクトメソッドが既に存在する場合、メソッド名コンボボックスからそのメソッドを選択します。
   または
   プロジェクトメソッドがまだ存在しない場合、メソッド名コンボボックスにメソッド名を入力し、**[...]**ボタンをクリックします。
   後者の場合、4Dはメソッド作成ダイアログボックスを表示します。
   **注:** メニューで使用されているメソッドの名前を変更した場合、メニューバーエディターで、添付されているメソッド名も編集する必要があります。
3. **新規プロセス開始**チェックボックスを選択する（任意）。
   **新規プロセス開始**チェックボックスを選択した場合は、このメニューコマンドを選択すると新規プロセスが作成されます。
   通常、メニューコマンドに割り当てたメソッドは、明示的にプログラムから新規プロセスを作成しない限り、カレントプロセスで実行されます。**新規プロセス開始**チェックボックスを選択すると、新規プロセスを簡単に開始することができます。
   **新規プロセス開始**チェックボックスを選択した場合、4D はそのメニューコマンドが選択されると新しいプロセスを作成します。プロセスリストにおいて、4D は"ML_ プロセス番号"というフォーマットでデフォルト名を新規プロセスに割り当てます。このように、メニューから開始されたプロセスの名前は、接頭辞"ML_"とプロセス番号を組み合わせて設定されます。プロセスに関する詳細は4Dランゲージリファレンスマニュアルのプロセスを参照してください。

### 標準アクションを割り当てる

メニューコマンドに標準アクションを割り当てるには:

1. メニューコマンドを作成するか、または選択する。
   プロパティエリアが変わり、選択したメニューコマンドのプロパティが表示されます。
2. 割り当てたいアクションを標準アクションメニューから選択する。
   メニューとして提供される標準アクションのリストは、ボタン用のものとほぼ同じです（4D のフォームエディターでボタンのフォームエディターから利用できます）。実際、大半のアクションは両方の場合で使用可能です。自動スプリッターアクションのみメニューコマンドに割り当てることができません。なのでこれはポップアップメニューに表示されません。各標準アクションについては標準アクションを参照してください。
   **Mac OSに関する注記:** Mac OS の場合、プラットフォームインタフェース標準に合わせるために、**環境設定**や**終了**アクションが割り当てられたカスタムメニューコマンドは自動的にアプリケーションメニュー内に置かれます。
3. (オプション) **イベントを発生させない**オプションを選択する。
   このオプションを選択すると、4D は関連アクションを実行する前に、カーソルが置かれているフィールドの"確認（バリデート）"を行いません。このオプションは主として**編集**メニューコマンドに使用されます。デフォルトとして、4D はフィールド内容を処理し、"確認"を行ってから、標準アクションを実行します（メニューコマンドやショートカットを使用）。これにより、On Data Changeフォームイベントが生成されます。しかし、コピー＆ペーストタイプのコマンドの場合、コマンドが呼び出されるとOn Data Changeフォームイベントが予想外に生成されてしまうため、その処理に差し支える可能性があります。その場合は、**イベントを発生させない**オプションを選択すると有効です。

## メニュープロパティを設定する

メニューエディターウィンドウの右では、フォントスタイルや区切り線、キーボードショートカット、ツールバーアイコンなど様々なメニュー項目プロパティを設定できます。

### カスタムパラメーター

各メニュー項目にカスタムパラメーターを関連付けることができます。メニュー項目パラメーターはその内容を自由に設定できる文字列です。

メニュー項目パラメーターは特にDynamic pop up menu、Get menu item parameter、そしてGet selected menu item parameter コマンドを使用する際、主にメニューをプログラムで管理するために利用できます。

### 区切り線

メニュー中のメニューコマンドグループは区切り線を使用して分割できます。この表示方法は機能ごとにメニューコマンドをグループ化するのに便利です。

区切り線を追加するには、メニューコマンドを作成します。カレントメニュー項目エリアにメニューコマンドのテキストを入力する代わりに、**区切り線**オプションを選択します。するとカレントメニュー項目に線が引かれます。

**注:**

- Mac OSではメニュー項目タイトルの一文字目を"-"にすると、その行が区切り線になります。この記法は特にINSERT MENU ITEMで有用です。
- **区切り線**オプションを選択すると、他のプロパティに効果はなくなります。

### キーボードショートカットを割り当てる

メニューコマンドにはショートカットを割り当てることができます。メニューコマンドにキーボードショートカットが割り当てられると、メニューを開いたときにそれがメニューコマンドの右に表示されます。例えば"Ctrl+C" (Windows) または ⌘+C" (Mac OS) が**編集**メニューの**コピー**コマンドの右に表示されます。

ショートカットには**Shift**や**Alt** (Windows) または **Option** (Mac OS) キーを追加できます。これにより使用できるショートカットの数を増やすことができます。以下のタイプのショートカットを定義できます:

- Windows:
  - Ctrl+文字
  - Ctrl+Shift+文字
  - Ctrl+Alt+文字
  - Ctrl+Shift+Alt+文字
- Mac OS:
  - Command+文字
  - Command+Shift+文字
  - Command+Option+文字
  - Command+Shift+Option+文字

**注:** 標準アクションに割り当てられたデフォルトのキーボードショートカットは変更しないことをお勧めします。

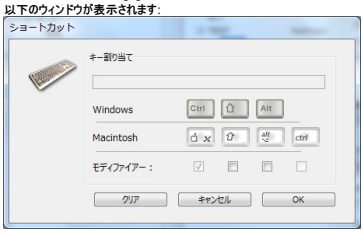
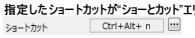
**ファイル**や**編集**メニュー、および4Dのメニューコマンドで使用されている標準メニューのショートカットを除き、すべての英数字をキーボードショートカット文字として使用できます。

予約されている組み合わせは以下の通りです:

| キー | 処理 |
|---|---|
| Ctrl+C | コピー |
| Ctrl+Q | 終了 |
| Ctrl+V | ペースト |
| Ctrl+X | カット |
| Ctrl+Z | 取り消し |
| Ctrl+. (period) | 実行停止 |

*Mac OSではCtrlの代わりにCommandが使用されます。

キーボードショートカットを割り当てるには:

1. キーボードショートカットを割り当てるメニュー項目を選択します。
2. ショートカット入力エリアの**[...]**ボタンをクリックします。
   以下のウィンドウが表示されます:
   
   
3. 文字を入力し、(必要であれば) **Shift** そして **Alt** (**Option**) オプションを選択します。
   **注:**
   - 指定する組み合わせのキーを押すと、押したキーがウィンドウに反映されます (ただし **Ctrl/Command** キーは押しません)。
   - **Ctrl/Command** キーの選択を解除することはできません。このキーは必須です。
   内容を消去するには**クリア**をクリックします。
4. **OK**をクリックしてウィンドウを閉じます。
   指定したショートカットが"ショートカット"エリアに表示されます:
   
   

**注:** アクティブオブジェクトにキーボードショートカットを割り当てることもできます。Ctrl/Commandキーの割り当てが衝突した場合、アクティブオブジェクトが優先されます。アクティブオブジェクトへのキーボードショートカット割り当てについてはキーボードショートカットを参照してください。

### 有効にされた項目

メニュー項目を有効で表示するか無効で表示するか選択できます。ユーザーは有効なメニュー項目を選択できます。無効なメニュー項目は灰色で表示され、選択することはできません。

明示的に設定しない限り、4Dは自動でカスタムメニューに追加された項目を有効にします。例えば特定の条件下でENABLE MENU ITEMやDISABLE MENU ITEMを使用して有効化するために、初期状態を無効にすることができます。

**選択可**チェックボックスの選択が解除されていると、メニューコマンドは灰色で表示され、選択することができません。

### チェックマーク

このオプションを使用して、メニュー項目にシステムチェックマークを関連付けることができます。その後チェックマークの表示をランゲージコマンド (SET MENU ITEM MARKやGet menu item mark) で制御できます。

通常チェックマークは連続したアクションを行うメニュー項目に付けられ、そのアクションを現在実行中であることを示すために使用されます:

### フォントスタイル

メニューコマンドにフォントスタイル (太字、下線、イタリック) を適用することができます。

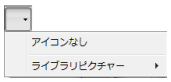
一般的なルールとして、フォントスタイルの適用は慎重に行ってください。煩雑なスタイルの使用はユーザーの注意をそらし、アプリケーションの見た目を悪くします。

スタイルを適用するにはメニューコマンドを選択し、チェックボックスを使用してスタイルを選択します。

### 項目アイコン

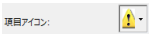
**項目アイコン**オプションを使用して選択した項目にアイコンを関連付けることができます。使用するアイコンは事前に4Dのピクチャーライブラリに格納し

ておかなければなりません。このオプションのエリアをクリックすると階層ポップアップメニューが表示され、ピクチャーライブラリからピクチャーを選択できます:




**注:** ピクチャーライブラリが空の場合、ピクチャライブラリメニューは表示されません。

選択するとプレビューエリアにアイコンが表示されます:




設定されたアイコンはメニューの左に表示されます:

項目からアイコンを取り除くには項目アイコンエリアのメニューから**アイコンなし**を選択します。

**ツールバーアイコン**

メニュー項目にツールバーアイコンを関連付けることができます。これを行うとアプリケーションモードのメニューバーに関連付けられたツールバーのボタンとしてそのアイコンが使用されます。

使用するアイコンは事前に4Dピクチャーライブラリに格納されていなければなりません。アイコンピクチャーの選択はメニュー項目アイコンと同様に行います (前の節参照)。

アイコンは20 x 20ピクセルのフレームで表示されます。4Dは自動でメニューコマンドのテキストをツールバーボタンのヘルプTipとして使用します。

**注:** アプリケーションモードのツールバーの表示/非表示はデータベース設定で設定します (参照)。

メニューコマンドに割り当てたツールバーアイコンを削除するには、ツールバーアイコンエリアのメニューから**アイコンなし**を選択します。

## メニューバーの管理

各メニューバーにカスタムスプラッシュスクリーンを関連付けることができます。またメニューバーとスプラッシュスクリーンをプレビューできます。

### スプラッシュスクリーンを設定する

各メニューバーにカスタムスラッシュスクリーンを関連付けることにより、アピアランスを拡張できます。スプラッシュスクリーンを含むウィンドウは、メニューバーが表示されるとき、その下に表示されます。ロゴなどのピクチャーを表示できます。デフォルトで4Dはスプラッシュスクリーンに4Dロゴを表示します:

任意の画像編集アプリケーションで作成したピクチャーをスプラッシュスクリーンで使用できます。ピクチャーにコピーした画像、ピクチャーライブラリ、あるいはハードディスク上の画像を使用できます。4Dがサポートする標準のピクチャータイプの画像を使用できます。

スプラッシュスクリーンピクチャーを変更するには:

1. メニューエディターでカスタムスプラッシュスクリーンを割り当てたいメニューバーを選択します。
   ウィンドウ右側に"背景画像"エリアが表示されます。
2. ディスクに保存されたピクチャーを直接開くには、開くボタンをクリックします。
   または
   "背景画像"エリアをクリックします。
   ピクチャーを追加するための様々なオプションを提供するポップアップメニューが表示されます。
   - クリップボードのピクチャーをペーストするには**ペースト**を選択します。
   - ピクチャーライブラリに格納されたピクチャーを選択するには、**ライブラリピクチャー**を選択します (ピクチャーライブラリが空の場合、このメニューは表示されません)。
   - ディスクファイルとして保存された画像を開くには**開く**を選択します。
3. 開くを選択すると、標準のファイルを開くダイアログボックスが表示されます。使用するピクチャーを選択します。
   選択した画像がプレビューとして表示されます。

メニューバーをテストすると、設定の結果を見ることができます。アプリケーションモードではピクチャーはスプラッシュスクリーンに"トランケート (中央合わせ)"で表示されます。

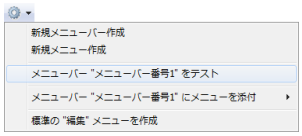
**注:** データベース設定を使用してスプラッシュスクリーンの表示/非表示を設定できます (インターフェースページの"ユーザー"参照)。

カスタムピクチャーを削除してデフォルトに戻すには、**クリア**ボタンをクリックするか、"背景画像"エリアをクリックして表示されるメニューから**クリア**を選択します。

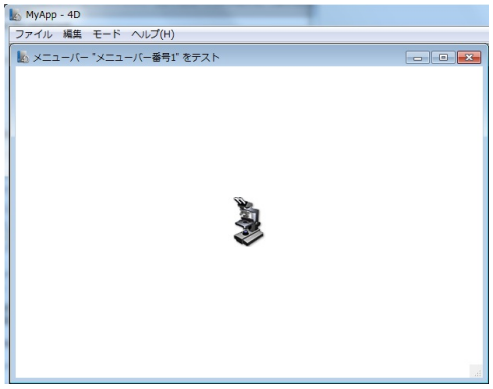
### メニューバーのプレビュー

メニューバーエディターからカスタムメニューとスプラッシュスクリーンをプレビューできます。アプリケーションモードに移動する必要はありません。

これを行うにはメニューバーを選択して、コンテキストメニューまたはエディターのオプションメニューから、**メニューバー "メニューバー #X" をテスト**を選択します。




メニューバーとスプラッシュスクリーンのプレビューが表示されます。




このプレビューではカスタムメニューに関連付けられているかもしれないツールバーは表示されません。
メニューや階層メニューを表示させることができますが、選択してもコマンドは実行されません。メニューやツールバーの動作を確認するには、**実行**メニューから**アプリケーションテスト**を選択します。

### メニューとカスタムアプリケーション

メニューバーはカスタムアプリケーションにおいて主要なインターフェースを提供します。各カスタムアプリケーションにおいて、最低一つのメニューを添付したメニューバーをひとつ作成しなければなりません。カスタムアプリケーションの作成については4Dランゲージリファレンスマニュアルの4Dアプリケーションのビルドを参照してください。

カスタムアプリケーションを作成するのか、アプリケーションモードで使用するメニューを作成するのかに関わらず、メニューバーを作成することができます。デフォルトでメニューバー番号1がアプリケーションモードで使用されます。SET MENU BARコマンドを使用して、メニューバーを変更することができます。

メニューコマンドにメソッドを割り当てない場合、そのメニューコマンドを選択するとアプリケーションモードが終了され、デザインモードに入ります (このモードへのアクセスが許可されていれば)。4D Desktopでアプリケーションを使用している場合、アプリケーションモードを終了するとアプリケーションが終了します。

デザインモードに入ることのできる4Dを使用している場合、デザインモードに入ることのできるユーザーを限定するためにパスワードアクセスシステムを使用します。データベース設定を使用して特定のグループにデザインモードへのアクセス権を割り当てます (WA SET PREFERENCE参照)。
デザインモードへのアクセス権がないグループに属するユーザーは、メソッドなしのメニューコマンドを選択しても、**Alt+Shift+right クリック** (Windows) または **Control+Option+Command+クリック** (Mac OS) ショートカット (このショートカットはプロセスポップアップメニューを表示します、During参照) を使用しても、アプリケーションモードには移動できません。デザインモードへのアクセス権がないユーザーがデザインモードに入ろうとすると、4Dは終了します。DesignerとAdministratorは常にデザインモードへのアクセス権があります。デザインモードへのアクセス権を持つグループに属する必要はありません。詳細はユーザー＆グループを参照してください。

# ピクチャーライブラリ


概要

ピクチャーをライブラリに追加する

ピクチャープロパティ

サムネールを作成する

ライブラリからのドラッグ＆ドロップ

## 概要

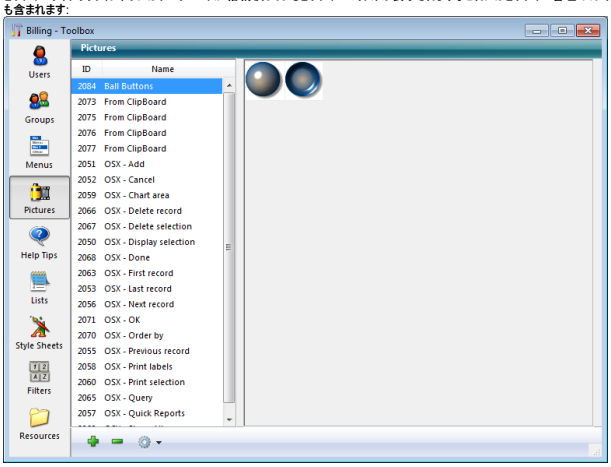
ピクチャーライブラリを使用してグラフィックを保存し、それらのピクチャーをフォーム、ツールバー、リストアイコン、ピクチャーメニュー項目または ピクチャーボタンなどのデザイン要素として使用することができます。ピクチャーライブラリを使用するとデータベースの複数の場所でグラフィックを使用でき ますが、その保存は 1 箇所だけで構いません。ピクチャーライブラリ内の画像を更新すると、その画像への参照が自動的に更新されます。この機能により、ストラクチャーファイルの サイズが小さくなり、データベースへの変更の管理がより簡単になります。

4Dはほとんどのカレントのピクチャーフォーマット (特にJPEG, SVG, PNG, BMP, GIF および TIFFフォーマット) をサポートしています。ピクチャーライブラリでは、ピクチャーは元のフォーマットで格納されます。フォーマットによって異なる特定の機能 (透過など) はそのまま保持され、変更されずに表示されます。[#cmd id="992"/]コマンドを使用して、そのマシンで使用できるネイティブフォーマットのリストを取得できます。

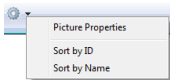
ピクチャーライブラリーには、ピクチャーボタンまたはピクチャーメニューを作成するためにサムネールのテーブルを作成・変更する、統合された機能が含まれています。

ピクチャーライブラリは4Dのツールボックスにあります。ピクチャーライブラリを表示するには:

1. **デザイン**メニューから**ツールボックス > ピクチャーライブラリ**を選択します。
   または
   4Dツールバーのツールボックスボタンをクリックし、ピクチャーページを表示します。
   ピクチャーライブラリウィンドウにはデータベースに格納されているピクチャーのリストが表示されます。これにはピクチャー管理コマンドと、表示エリアも含まれます:




リストのコンテキストメニューまたはオプションボタンに関連付けられたコマンドを使用して、ピクチャーを名前順(デフォルト)またはID番号順に並べ替えることができます:



## ピクチャーをライブラリに追加する

### ピクチャーファイルを読み込む

4DはJPEG, SVG, PNG, BMP, GIF そして TIFFなど、ほとんどのピクチャーフォーマットを読み込んで表示することができます。

新しいピクチャーをピクチャーライブラリに読み込むには:

1. ピクチャーリストの下にある追加ボタンをクリックするか、ピクチャリストエリアを右クリックして**追加**を選択して下さい。
   標準のファイルを開くダイアログボックスが表示されます。
   「ファイルタイプ」のメニューをクリックすると4Dがサポートしているファイルフォーマットが表示されます。
2. 取り込むファイルを選択するか「**開く**」をクリックします。

または、

1. 外部のピクチャファイルをライブラリーのピクチャリストの中にドラッグ&ドロップします。

ピクチャが有効な物であればピクチャプロパティダイアログが表示されます。画像のサイズはインポートされたピクチャーに応じて自動的に定義されます。
詳細に関しては、ピクチャープロパティを参照して下さい。

必要であれば、ファイル名・ID番号や他のプロパティを変更して下さい。**OK**をクリックするとピクチャが作成されます。

**警告:** ピクチャライブラリの中で作成されたピクチャに関してはID番号のみ変更することが出来ます。

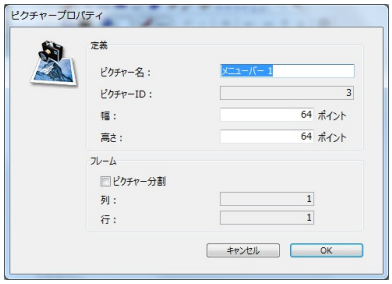
### ピクチャの削除

ライブラリーからピクチャーを削除するには以下の手順に従って下さい:

1. ピクチャーリストの中から削除したいピクチャーを選択し、削除ボタンをクリックして下さい。
   または
   ピクチャーを右クリックして、コンテキストメニューから**削除**を選択して下さい。

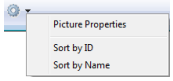
## ピクチャープロパティ

「ピクチャープロパティ」ダイアログボックスを使用すると、ピクチャーの名前やサイズ、フレームの各属性を設定したり、表示することができます。またID番号も表示されますが、ピクチャー追加時を除き変更することはできません。




ピクチャーの追加方法には関係なく、上ようなダイアログボックスが表示されます。

このダイアログを表示するには、ピクチャーをダブルクリックするか、ピクチャーを選択してからライブラリのオプションメニューで**ピクチャープロパティ**を選択します。




このダイアログボックスには次のプロパティが表示されます:

- **ピクチャー名**: 複数のピクチャーに同じ名前を指定できますが、ID 番号だけは異なるものにしなければなりません。
- **ピクチャーID**: ピクチャーのユニークなID 番号。この番号はそのピクチャーの参照番号です。ピクチャーボタンやピクチャーポップアップメニュー、カスタムツールバー、リストを作成する場合や、プログラムからピクチャーを操作する場合は、この番号を使用してピクチャーを参照します。
  **注:** ピクチャーの作成時にこの番号を定義することができますが、後で変更することができません。
- **幅**および**高さ**: ピクチャーのサイズ。これらの値は、ピクチャーを読み込む際に（ファイルまたはクリップボードから）事前に算出されます。ピクチャーを分割した場合（後述）、この値は各フレームのサイズに相当します。
- **フレームエリア**: ひとつの画像をもとにして、ボタン配列やピクチャーメニューの作成に使用するサムネールを作成することができます。詳細はサムネールを作成するを参照してください。

## サムネールを作成する

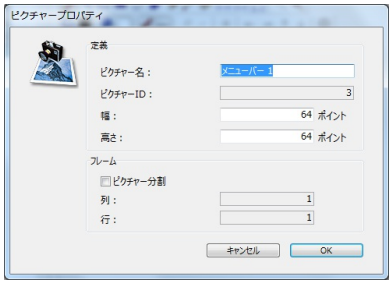
ピクチャーライブラリには、ピクチャーボタンやピクチャーメニューで使用するピクチャーを作成するために、行と列の配列でピクチャーを作成/修正できる機能が用意されています。一次元または二次元の配列を作成できます。ピクチャーライブラリでは、このような配列を"フレーム"と呼びます。4D の他の場所では、ピクチャー配列をピクチャーの"サムネール"または"テーブル"と呼びます。この目的は、行と列を使用してピクチャーを分割することにあります。その結果、各セルは"フレーム"または"サムネール"とみなされます。4D は指定されたパラメータに従い、正しいフレームをピクチャーボタンやピクチャーメニューに表示します (詳細はピクチャーボタンとピクチャーポップアップメニューを参照してください)。

ピクチャーの作成時、または作成後でも、フレームのシーケンス(連続)を定義することができます。

### フレームのシーケンスを作成する

フレームシーケンスは、ピクチャーライブラリに格納されたピクチャーを元に作成するか、またはピクチャーライブラリへピクチャーを追加する際に作成することができます。いずれの場合でも、ピクチャープロパティダイアログボックスにおいて、ピクチャーを別々のフレームに分割する方法を指定することができます。ピクチャーを作成している場合は、ダイアログボックスが自動的に表示されます。それ以外の場合は、ピクチャーをダブルクリックするか選択して、ライブラリのオプションメニューから**ピクチャープロパティ**を選択します。

フレームエリアを使用し、作成するフレームシーケンスの行数と列数を定義することができます。フレームを作成するには、まず初めに**ピクチャー分割**オプションを選択しなければなりません:




### フレームのサイズ

フレームのサイズは、4D により自動的に算出されます。フレームのシーケンスを定義する際に、幅と高さエリアが変更されて各フレームのサイズが表示されます。

フレームのサイズを後から変更したい場合は、ピクチャーの全体サイズは気にせずに、幅と高さエリアに新しい値を入力します。新しいサイズが前より大きい場合は、変更後の各フレームは新しいサイズで中央に寄せられます（ピクチャーが歪むことはありません）。新しいサイズが前より小さい場合、各フレームは切り捨てられます。

## ライブラリからのドラッグ & ドロップ

ピクチャーをフレームシーケンスとして定義すると、ショートカットキーを使用して、それをピクチャーボタンやピクチャーポップアップメニューとしてフォームに挿入することができます。

- ピクチャーボタンを作成するには、ライブラリからピクチャーをドラッグしてフォームへドロップします。
- ピクチャーポップアップメニューを作成するには、**Shift** キーを押しながらライブラリからピクチャーをドラッグして、フォームへドロップします。
- 標準のピクチャーを作成するには、**Alt** (Windows) または **Option** (Mac OS) キーを押しながらライブラリからピクチャーをドラッグして、フォームへドロップします。

フレームシーケンスとして定義されていないピクチャーをドラッグ&ドロップすると、標準のピクチャーとして挿入されます。

**注：** フォームエディターでプロパティリストを使用し、ピクチャーライブラリ（ライブラリ内のソースピクチャーが変更されるとダイナミックに更新される）と、スタティックピクチャー（ライブラリ内のソースピクチャーに関連付けられていない）という2 種類のピクチャーを区別することができます (スタティックピクチャーを使用する参照)。

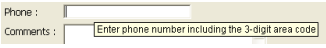
## ヘルプTips


概要

ヘルプTipエディター

## 概要

フォーム上のフィールドやアクティブオブジェクトにヘルプTip を付加すると、データベースをより生産的に操作できるようになります。ヘルプメッセージはマウスがオブジェクトの上にあるとき、Tipとして表示されます。



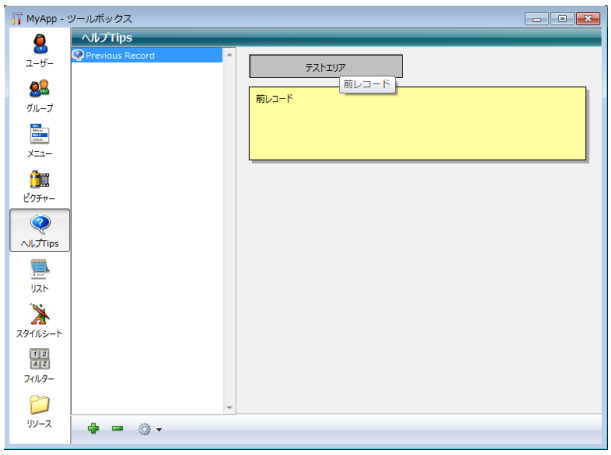

オブジェクトにヘルプメッセージを関連付けるにはプロパティリストに直接入力するか、ヘルプTipエディターで作成したヘルプTip参照を挿入します。

データベースオブジェクトにヘルプTipを関連付ける方法に関する詳細は一般ページの"ヘルプメッセージ"を参照してください。

## ヘルプTipエディター

ヘルプTipエディターを使用してヘルプTipを作成・テストすることができます。作成したヘルプTipはデータベースオブジェクトに関連付けることができます。

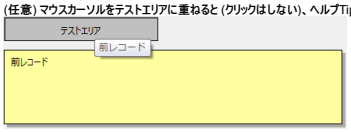
このエディターを表示するには、4Dツールボックスの**ヘルプTip**ボタンをクリックします:




- 左のリストには作成済みヘルプTipの名前がリストされます。このリストにはデフォルトフォームやフォームウィザードを使用してフォームを作成したときに自動で生成されるヘルプTipも含まれています。
- グレーのエリアにマウスを重ねると、ヘルプTipをテストできます。
- 黄色のエリアに、表示したいメッセージを入力します。

### ヘルプTipを作成する

ヘルプTipエディターを使用してヘルプTipを作成するには:

1. 追加ボタンをクリックします。
   "ヘルプ #1"という名前の項目がリストに追加されます。
2. (任意) 名前を変更します。
   名前はアプリケーションからヘルプTipを参照するために使用されます。名前はユーザーに表示されません。15文字まで使用できます。
   **Alt** (Windows) または **Option** (Mac OS) キーを押しながらヘルプTip名をクリックすると、名前編集モードになります。またはオプションメニューから**名前を変更**コマンドを選択することもできます。
   デフォルト名を変更しない場合、次に作成するヘルプTipの名前はヘルプ#2、ヘルプ#3...となります。
   **注**: プロパティリストに直接入力されたヘルプTipはこのエディターに追加されません。
   名前は文字順に表示されます。新しいヘルプTipを追加すると、リストの内容が自動で更新されます。
3. **Tab**キーを押すか入力エリアをクリックして、ヘルプTipのメッセージを入力します。
   255文字まで使用でき、XLIFF参照を指定することもできます (スタティックテキスト中で参照を使用する参照)。
   **注**: フィールドや変数タイプの動的参照はエディターのテストエリアに表示されません。
4. (任意) マウスカーソルをテストエリアに重ねると (クリックはしない)、ヘルプTipの表示をテストできます。
   
   
   データベース実行中に表示される通りに、プレビューすることができます。
5. 既存のヘルプTipの複製を作成したい場合、元のヘルプをリストから選択してオプションメニューから複製を選択します。

### ヘルプTipを更新/削除する

ヘルプTipを選択して内容を変更できます。メッセージを変更して、**Tab**キーを押すか他のエリアをクリックすると、変更が保存されます。

ヘルプTipを削除するには、それを選択して削除ボタンをクリックします。

# リスト


概要

データ入力用のリストを設計する

リストの作成と変更

リストプロパティの設定

リストエディターからのドラッグ＆ドロップ

## 概要

リストはフィールドや入力可能オブジェクトで選択肢として提供される一連の値です。リストは以下の用途で利用できます:

- フィールドや入力可能オブジェクトでユーザーに選択肢を提供する。
- 入力可能な値をリスト項目に制限する。
- 入力される値からリストの項目を除外する。

階層リストを作成することもできます。階層リストはリスト項目に別のリストを関連付けたものです。

リストや階層リストの各項目に小さなアイコンを関連付けることができます。必要に応じてこのアイコンは項目の左側に表示されます。例えばスクロールエリア (ポップアップメニュー、ドロップダウンリスト参照)、タブコントロール、そして階層リスト (階層ポップアップメニューと階層リスト参照) に小さなアイコンを表示できます。

フィールドや入力可能オブジェクトの選択リストとしてリストが使用されると、ユーザーは値を入力する代わりに、このリストから値を選択することができます。例えば、人事データベースに役職名を入力するために選択リストを作成したいかもしれません。

また、リストを使用してデータ入力に制限を設けることもできます。あるリストではフィールドの指定値を設定して、それ以外の値をすべて除外し、別のリストではフィールドの除外値を設定して、リスト内の値が入力されないようにすることもできます。

1 つのデータベース内でリストには8,000 項目まで設定することができ、各項目には20億桁まで格納することができます。

フィールド属性としてフィールドに選択リストを設定する方法についてはフィールドプロパティを参照してください。データ入力制御のためにリストを使用する方法についてはデータ入力の制御と補助を参照してください。

多くの場合、リストはメソッド内で使用されます。例えば、配列要素をリストに格納すると便利です。配列は、値のリストをメモリ上に保存します。リストを使用して、ポップアップメニュー、階層リスト、コンボボックス、タブコントロール、および他の複数値によるインタフェースオブジェクトの各要素を保存することができます。メソッドを使用するか、プロパティリストウインドウでオブジェクトにリストを割り当てることにより、リスト内容をインタフェースオブジェクトに転送します。

リストは、ツールボックスのリストエディターで作成します。

**4D Serverに関する注記:** 2人以上のユーザーが同時に同じリストを編集しようとすると、オブジェクトのロックが発生します。デザインモードであるユーザーが編集中のリストはロックされます。他のユーザーは最初のユーザーがリストを閉じてロックを解除するまで、そのリスト、リスト名、リスト項目を編集できません。

## データ入力用のリストを設計する

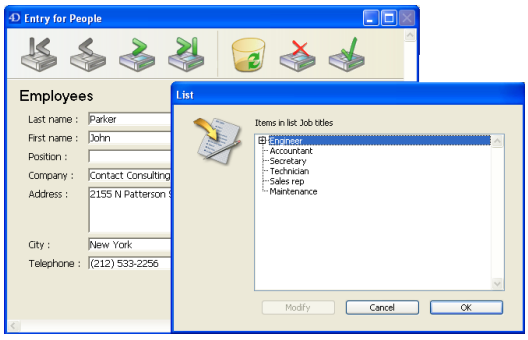
リストの用途のひとつは、データ入力時に値を選択するために値一覧をユーザーに提供することです。この目的をはじめとして、リストを使用する理由をいくつか次に示します:

- あらゆるフォーム、または選択したフォームでリストを共有できます。
- 長いリストは複数の小さなリストに分割できます。
- 入力できる値をリストの項目に制限したり、またはユーザーに別の値の入力を許可することができます。
- ユーザーにリストの変更を許可したり、またはリストの変更を禁止することができます。
- ポップアップメニュー/ドロップダウンリストまたはコンボボックスを使用することによって、フィールドや変数を直接フォームに表示させることができます。

**Tip:** リストに含まれる項目の数が限定されている場合、リストは必要ないかもしれません。例えば、二つしか値のないリスト(男性/女性)の場合、ラジオボタンまたはピクチャーラジオボタンをチェックすることによってデータを入力可能なプール型のフィールドを使用した方が適しているかもしれません。異なる値が3つや4つだったとしても、チェックボックスを使った方が適切な場合もあります。

フィールドプロパティあるいはデータベースストラクチャーレベルでリストをフィールドに割り当てることができます。フィールドにリストを割り当てると、リストフォームか詳細フォームかに関わらず、そのフィールドをデータ入力のために選択すると、常にウィンドウ内にリストが表示されます。ユーザーはこのリストから値を選択することができます。リストが並べ替えられている場合は、ユーザーがキーボードで文字を入力すると、リストが自動的にスクロールされます。例えば、"N"と入力すると、リストは"N"で始まる最初の項目までスクロールされます。ユーザーは目的の項目が現れた時点で入力を中止し、その項目をリストから選択することができます。

次の図は、選択リストが表示されている様子を示しています。




ストラクチャエディターのインスペクタを使用してフィールドにリストを割り当てた場合は、クエリエディターでそのフィールドが選択された際にもリストが表示されます。詳細は フィールドプロパティ の「項目選択を使用する」を参照してください。

また、フォームのデータ入力制御としてリストをフィールドに割り当てることもできます。この場合、リストはその入力フォームでフィールドが選択された場合にのみ表示されます。

この場合、ポップアップメニュー、ドロップダウンリスト (変更不可能な選択項目)または コンボボックス (変更可能な選択項目)を使用することによって、選択リストダイアログボックスを使用することなくフィールド内の入力と表示を管理することができます。これのためには、オブジェクトをリストに割り当て、プロパティリストの「変数名」にフィールドまたは変数名を入力するだけです。詳細な情報についてはポップアップメニュー、ドロップダウンリストを参照して下さい。

### 階層リスト

4Dでは階層リストを作成できます。親リストを選択するとサブリストが表示されます。

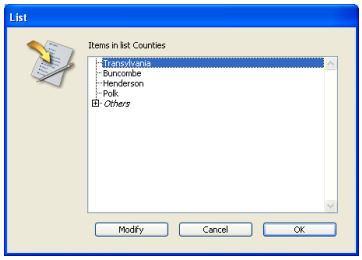
階層リストの利用方法には標準といえるものがなく、その使用目的により変わります。例えば、都市リストには項目が数多くあるため、選択に時間がかかります。このような問題のために階層リストを使用したい場合は、いくつかの対処法があります。

入力のためにリストを使用する以外にも、階層リストを使用して次のオブジェクトに値を代入することができます:

- タブコントロール,
- 階層ポップアップメニューと階層リスト,
- リストボックス.

**注:** 階層リストは、**ポップアップメニュー/ドロップダウンリスト**または**コンボボックス**に関連付けることが可能です。ただしこの場合、階層リストの一番上の階層のみが表示されます。

値のリストを2種類の値に分けることがよくあります。前述の例では、最も頻繁に選ばれた都道府県を割り出してみます。選択された値のうち80%が一部の都道府県であれば、これらの都道府県をリストに入れ、残りの都道府県を次の図のようなサブリストに格納することができます。




これにより、全体の80%の項目を直接選択できるようになり、多くの場合で入力処理が速くなります。

これとは別に、地域別に都道府県をまとめる方法もあります。最初に表示されるリストから地域を選択して、サブリストにより都道府県を選択することができます。この場合、地域を選択してから都道府県を選択する必要がありますが、47都道府県のリストの中から1つを選択するよりは速いでしょう。

### 指定値と除外値

データ入力作業のなかには、さほど重要ではないものもあります。リストに存在しない値を入力しても、まったく問題がないような場合です。しかし、リスト内に存在する値を入力しなければならないアプリケーションもあるかもしれません。別の値を入力してしまうと、請求書の支払遅延などの深刻な結果をもたらす場合もあるでしょう。

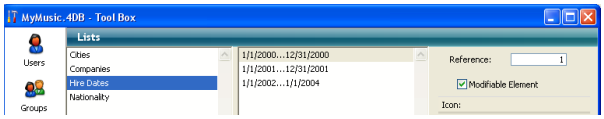
4D では、フォームのデータ入力制御の一部として指定値リストを設定することができます。このタイプのデータ入力制御を使用すると、ユーザはリスト上にある値以外の値を入力できなくなります。例えば、人事データベースにおいて、会社で許可される役職名が特定のものに限定されている場合です。

もう一つのデータ入力制御では、リスト内の値を除外することができます。この場合ユーザーはフィールドに格納すべきではない値を入力できません。例えば、会社では特定の国々で商売を行なわないかもしれません。これらの国々を除外リストに入れると、その値が入力できなくなります。

### 不連続な値の範囲

数値や日付、時間の各フィールドに設定する最大値と最小値は、最も効果的なデータ入力制御の1つです。最大値と最小値を設定すると、ユーザはこの範囲外の値を入力できなくなります。

フィールドに3種類の範囲を受け入れる場合について考えてみましょう。リストを使用して、このような不連続な範囲を設定することができます。このリストをフィールドの指定リストに設定し、この3つの範囲以外の値を受け入れないようにすることができます:



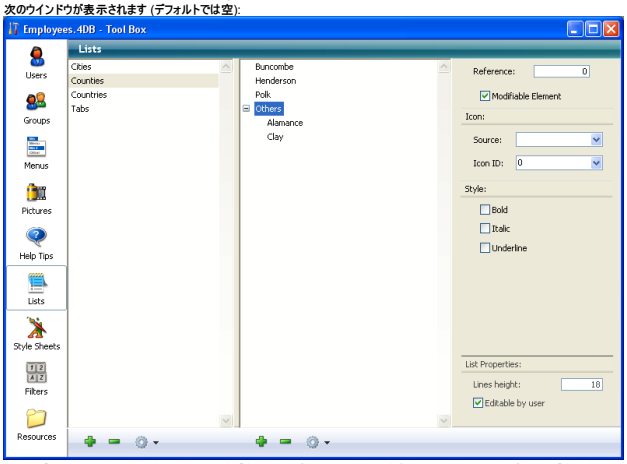

これとは逆に、有効ではない範囲を指定するリストを作成することもできます。フィールドの除外値リストとして、このようなリストを作成すると、その範囲内の入力値は受け入れられなくなります。

# リストの作成と変更

## リストを作成する

リストは、ツールボックスのリストエディターで作成します。また、リストエディターに戻って変更を行うことにより、いつでもリストを修正することができます。

リストを作成するには:

1. **デザイン**メニューから**ツールボックス>リスト**を選択します。
   次のウインドウが表示されます (デフォルトでは空):
   
   
   リストエディターの左側には、既存のリストの名前が表示されます。ウインドウの中央部分にはカレントリストの各項目が表示され、右側にはカレント項目とリストのプロパティがが表示されます。
2. リストの一覧の下にある追加ボタンをクリックします。
   または
   リストの一覧で右クリックし、コンテキストメニューから**追加**コマンドを選択します。
   4Dは"新規リストX"という名前の新規リストを作成します。

   
   **注:** ひとつでもリストが存在する場合、**複製**コマンドが利用可能になります。このコマンドを使用すれば既存のリストの性質を保持した新規リストを素早く作成できます。
3. リスト名を変更し、**Tab** キーを押して入力を確定します。
   コンテキストメニューの**名称変更**コマンドを選択するか、または**Alt+ クリック** (Windows) か**Option+ クリック** (Mac OS) ショートカットを使用、あるいは変更したいリストの名前を2 回クリックすると、いつでもリスト名を変更することができます。

これで新しい空のリストが作成されました。次は、このリストに表示される項目を作成します。

## リストに項目を追加する

リストに項目を追加する際、リストの最後に新しく項目を追加するか、あるいは既存リストの任意の場所に項目を挿入することができます。

項目をリストに追加するには:

1. 項目を追加したいリストの名前を選択します。
   リストに項目が既に存在する場合は、その項目がエディターの右側にリストとして表示されます。既存のリストに値を挿入したい場合は、挿入したい値の後にある値を選択します。その値の後に新しい値が作成されます。
2. 項目リストの下にある追加ボタンをクリックします。
   または
   リスト項目のコンテキストメニューから**追加**コマンドを選択します。

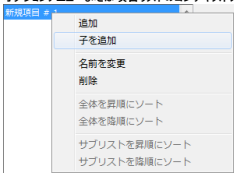
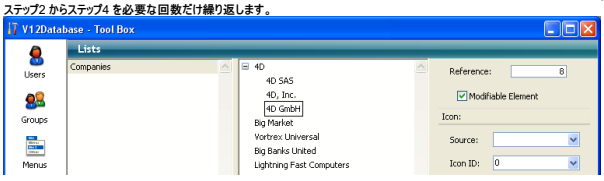
   
   4Dは"新規項目 # X"という名前の新規項目を作成します。
3. 項目の値を入力し、Tab キーを押して入力を確定します。
   コンテキストまたはオプションメニューの**名称変更**コマンドを選択、あるいは2回クリックして、いつでもリスト項目の値を変更することができます。
4. リストに他の項目を追加するには、ステップ2 とステップ3 を必要な回数だけ繰り返します。

   
   値を入力した後は、ドラッグ＆ドロップ操作によりリストの項目を移動することができます。また、アルファベット順に項目を並べ替えることも可能です (後述のリストを並べ替えるを参照してください) 。

## 階層リストを作成する

任意のリスト項目にリストを追加することができます。階層構造のレベルに仕様上の制限はありません。

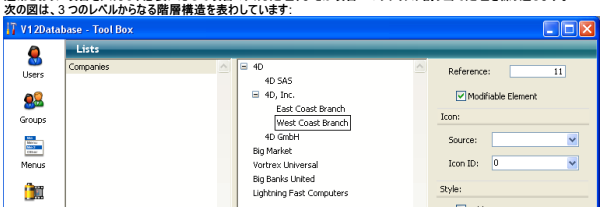
階層リスト作成するには:

1. サブリストを割り当てるリスト項目を選択します。
2. オプションメニューまたは項目リストのコンテキストメニューから**子を追加**を選択します。
   
   
   4Dは選択されたリスト要素を展開し、サブリスト上の最初の項目として新しい項目を作成します (デフォルトで"新規アイテム # X"という名前が付けられます) 。
3. 項目の値を入力します。
   コンテキストメニューやオプションメニューの**名称変更**コマンドを選択、または項目上を2回クリックすると、いつでもサブリスト項目の値を変更することができます。
4. サブリストに別の値を追加するには、その項目を選択したまま項目リスト下の追加ボタンをクリックします。
   または
   または親項目を選択し、**子を追加**コマンドをもう一度使用します。
   サブリストの項目が選択されている状態で**子を追加**コマンドを選ぶと、その階層構造内でより低いレベルが追加作成されます (後述) 。
5. ステップ2 からステップ4 を必要な回数だけ繰り返します。
   
   

必要があれば、サブリスト項目にサブリストを割り当てて、階層構造を続けることができます。

サブリスト項目にサブリストを割り当てるには:

1. サブリストの項目を選択します。
2. オプションメニューまたはリスト項目のコンテキストメニューから、**子を追加**を選択します。
3. 通常どおりに項目を入力し、必要に応じて項目の入力処理、または項目へのサブリスト割り当て処理を繰り返します。
   次の図は、3 つのレベルからなる階層構造を表わしています:




**リストで範囲を指定する**

4Dでは数値や日付、時間の範囲をリストに設定することができます。フォーム上でリストを指定値リストや除外値リストに設定すると、これらの範囲をデータ入力チェックに利用することができます。

リストに範囲を作成するには:

1. 範囲として使用したいリストを作成します。
2. 各項目に対して範囲の最小値、2 つのピリオド (..) 、最大値を入力します。
   例えば"100..150"は100 以上150 以下の範囲を設定します。
3. 必要な数を定義するまで範囲を別々の項目として指定します。
   範囲を使用したリストの例を次に示します:

**リストを並び替える**

4D では、項目リストは入力された順に並べられます。データベースユーザが項目をより利用しやすくなるように、リストやサブリストを文字順に並び替えることができます。リストを並び替えておくと、キーボードで入力された文字と一致するまでリストが自動的にスクロールされるため、データをさらに入力しやすくなります (日本語を除く)。

リストやサブリストを並べ替えるには:

1. 並び替えたい項目を含むリストを選択します。
2. コンテキストメニューあるいは項目リストの下にあるオプションメニューから、並び替えオプションを選択します。以下のオプションを利用可能です:
   - **すべてを昇順に並び替え**: 4D はリストやサブリストを昇順 (A ～ Z) に並び替えます。
   - **すべてを降順に並び替え**: 4D はリストやサブリストを降順 (Z ～ A) に並び替えます。
   - **サブリストを昇順に並び替え**: 4D はサブリストだけを昇順に並び替えます。
   - **サブリストを降順に並び替え**: 4D はサブリストだけを降順に並び替えます。

**項目とリストを削除する**

階層構造の任意のレベルで、項目を削除することができます。項目やリストの削除は取り消すことができないという点に留意してください。

項目を削除するには:

1. 削除したい項目を含むリストを選択します。
2. カレントリストエリアから削除したい項目を選択します。
   必要があれば、そのリストを展開します。
3. カレントリストの項目リストの下にある削除ボタンをクリックします。
   または
   リスト項目のコンテキストメニューから**削除**コマンドを選択します。
   4Dはリストから項目を削除します。

リストを削除するには:

1. 削除したいリストを選択します。
2. リスト一覧の下にある削除ボタンをクリックします。
   または
   リスト一覧のコンテキストメニューから**削除**コマンドを選択します。
   4Dはダイアログボックスを表示し、ここで操作の確認やキャンセルを行うことができます。このダイアログボックスを確定すると、リストが削除されます。

## リストプロパティの設定

### 項目に参照を追加する

カレント項目のプロパティエリアには、その項目参照を入力するためのエリアがあります。この参照は、その項目の固有IDとして設定されます。この参照は、メソッドを使用してリストを管理する場合、もしくはリストフィールドまたは変数においてリスト項目の参照番号オプションが有効化されている場合に使用されます(関連付け(値または参照番号)を参照して下さい)。

参照: 1

ランゲージを使用して、ユーザが選択したリスト項目 (例:階層メニュー内で選択された項目) を知る必要があれば、その項目の参照ID を利用して、ユーザが選択した項目を特定することができます。詳細は4D ランゲージリファレンスマニュアルの[#title id="95"/]を参照してください。

### 更新可能要素オプション

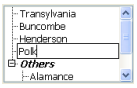
リストエディターではリストの各要素ごとに設定できる**更新可能要素**オプションがあります。このオプションはデフォルトで選択されています。

このオプションは階層リストやタブコントロールとして表示されるリストにのみ使用されます。

**注:** リストがコンボボックスに関連付けられているとき(コンボボックス を参照)、その要素は常に変更可能です。

#### 階層リスト

リストを階層リストオブジェクトの項目として指定することができます。この場合、各項目をユーザーが編集できるかどうかを指定することができます。あるリスト項目が編集可能である場合、**Alt** キー (Windows) または **Option** キー (Mac OS) を押しながらクリック (あるいは2回クリック) すると、その項目が編集モードになります。編集モードの階層リストは以下のようになります:




この場合、**更新可能要素**オプションは、ユーザーに階層リストの要素を変更することを許可します。

#### タブコントロール

タブコントロールにリストが関連付けられている場合、リストの項目に対応する各タブコントロールを有効/無効にできます。無効なタブコントロールはフォーム中でグレー表示されます。以下の例では"Henderson"が無効になっています:

この場合、**編集可能項目**オプションは、要素に対応するタブコントロールを有効にします。

### 項目にアイコンを設定する

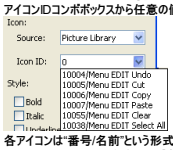
リスト上の項目にスモールアイコンを関連付けることができます。リストがスクロールエリアやタブ上に表示されると、その値の左側にこのアイコンが表示されます。

使用するスモールアイコンは、ピクチャーライブラリまたは'cicn'か'pict'タイプのリソース内に保存されます。

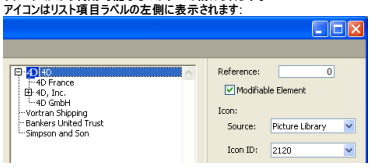
**互換性に関する注記:** Macリソースの利用は推奨されません。このテクノロジは廃止予定です (4Dランゲージリファレンスマニュアルのリソース参照)。これらのオプションは互換性のために保持されています。ピクチャーライブラリの利用を推奨します。

アイコンをリスト項目に割り当てるには:

1. リストを選択してから、アイコンを関連付けたい値を選択します。
2. アイコンエリアの**ソース**メニューから任意のオプションを選択します。
   このメニューを使用して、アイコンのソースを指定します (**ピクチャーライブラリ**オプションを推奨します)。
   オプションを選択すると、アイコンIDメニューの内容が更新され、データベースで利用できるピクチャーのうち、ソースメニューで指定したタイプに対応するピクチャーがすべて一覧されます。
3. アイコンIDコンボボックスから任意の値を選択します。




各アイコンは"番号/名前"という形式で表示されます。ソースメニューのリソースを選択すると、アイコンIDメニューには、該当するリソースのうちアプリケーションから利用可能なものがすべて納められます。
アイコンはリスト項目ラベルの左側に表示されます:



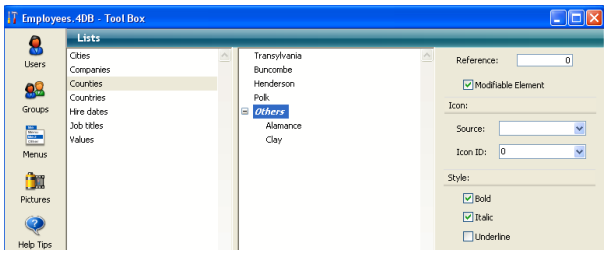

**注**: 使用するアイコンのサイズ次第では、リストの高さを変更する必要があるかもしれません。詳細は、後述のリストの最小の高さを設定するを参照してください。

アイコン参照を削除するには、アイコンIDコンボボックスの値を選択してから**Backspace** キーを押すか、または0 を入力します。

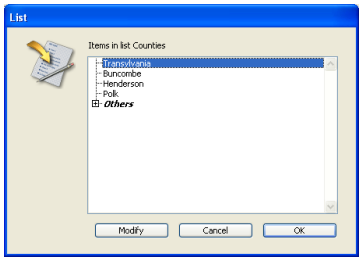
### フォント属性を指定する

選択リストとしてリストを使用する場合、リスト項目を太字、斜体、下線付きで表示することができます。

スタイルを適用するには、リストの値を選択してから、**太字**、**イタリック**、**下線**のうち希望するオプションをチェックします (フォントがサポートしないスタイルは使用できません)。スタイルを組み合わせて使用したい場合は、複数のオプションを選択することができます。次の図は太字とイタリックの属性がリスト項目に適用されている様子を示しています。




このリストを選択リストとして使用すると、次の図で示すように、選択したスタイル属性が用いられます:

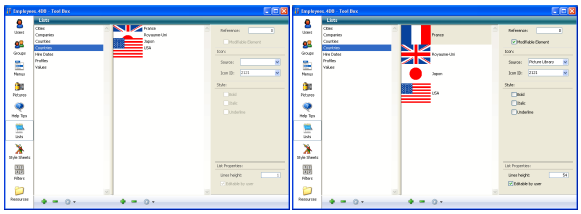


**リストの最小の高さを設定する**

4D が選択リストとしてリストを表示する場合は、階層リストオブジェクトのフォントサイズを使用して、隣接する各リスト項目の縦方向の間隔を決定します。階層リストの値を指定するためにリストを使用する場合は、縦の間隔をより大きく指定することができます。

このような操作を行う主な理由は、リスト項目に割り当てたアイコンのためにスペースを余分に用意するためです。あるいは、この機能をリストの各項目の間隔を拡げるためだけに利用することもできます。

最小の高さを指定するには、行高入力エリアに任意の値をポイント単位で入力します。
この値の結果は項目リストのエリアに即座に表示されます。次の図は、最小の高さの値を大きくした結果を表わしています。:

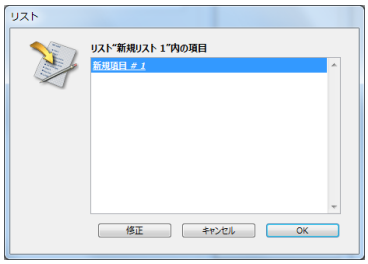


**選択リストをユーザー更新可に設定する**

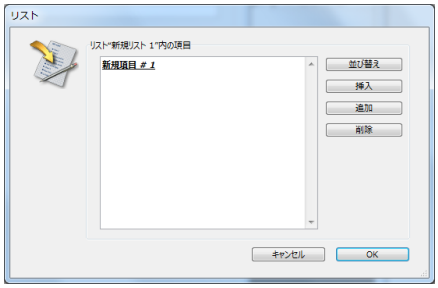
4D では、リストが選択リストとして表示された場合に、ユーザーが項目リストを変更できるかどうかを指定することができます。デフォルトでは、リストを変更できません。

リストをユーザー編集可に設定すると、ユーザーは特別なリストエディターを利用できるようになります。この特別なリストエディターでは、指定されたリストだけを編集できます。リストの追加や削除、別のリストの変更は行えません。リストが更新可に設定されている場合は、そのリストの項目を自由に変更することができます。

リストが更新可能な場合、ユーザーモードのリストダイアログボックスで**修正**ボタンを使用できるようになります。




ユーザーがこのボタンをクリックすると、リストエディターが表示されます:




リストをユーザー更新可能にするには:

1. 編集可能に設定したいリストを選択します。
2. リストプロパティエリアの**ユーザー更新可**オプションを選択します:

これで、リストをユーザーが修正できるようになります。

ユーザーがリストを変更できないようにするには、リストを選択して**ユーザー更新可**オプションの選択を解除します。

## リストエディターからのドラッグ & ドロップ

4D では、ショートカットを使用して、リストに割り当てる階層リストやドロップダウン階層メニューを作成することができます。リストエディターからリストをドラッグして、これらの処理を行えます:

これらのオブジェクトに関する詳細は階層ポップアップメニューと階層リストを参照してください

- 階層リストをフォームに挿入するには、リストエディターからフォームへリストをドラッグします。
- 階層メニューをフォームに挿入するには、**Shift** キーを押しながらリストエディターからフォームへリストをドラッグします。

# スタイルシート


概要

スタイルシートを作成する

スタイルシートを適用する

自動スタイルシート

## 概要

スタイルシートは一連のフォント属性 (フォント、フォントサイズ、書体) の組み合わせをグループ化したものです。作成したスタイルシートはフォームエディターでオブジェクトのフォント属性として設定できます。

各 スタイルシートは4Dがサポートする各プラットフォーム (Windows XP, Windows Vista/7, Windows Classic, Mac OS X) インターフェース用のフォント属性を個別に設定して保存します。例えば"Buttons"という名前のスタイルシートに、Mac OS用にLucida Grandeを、Windows用にTahomaフォントを設定できます。同様にフォントサイズも設定できます。

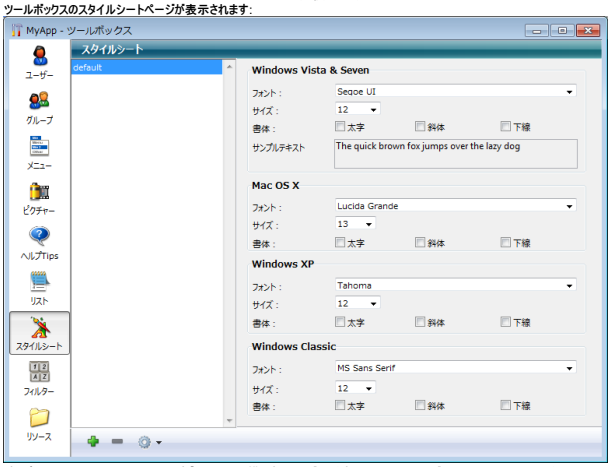
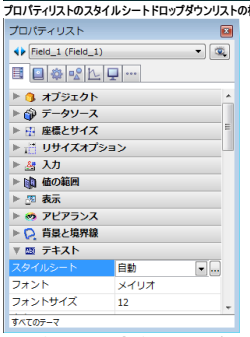
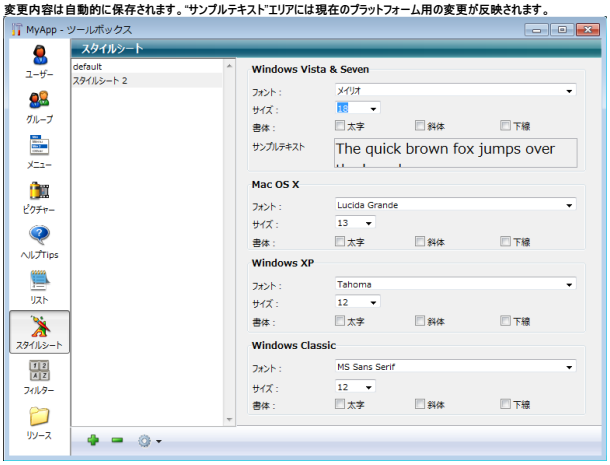
アプリケーションのインターフェースを調和させるのに加えスタイルシートに利用には3つの利点があります:

- 開発にかかる時間を少なくする: オブジェクトのフォント属性を一括で設定できます。
- メンテナンスが容易: スタイルシートはそれが使用されているすべてのオブジェクトの外観を更新します。例えばスタイルシートでフォントサイズを変更すれば、その変更は同じスタイルシートを使用するすべてのオブジェクトに適用されます。
- マルチプラットフォーム開発のコントロール: スタイルシートが適用されると、4Dは自動でフォームが表示されたプラットフォーム用に設定されたパラメーターを使用します (オブジェクトにシステムアピアランスが設定されている場合)。
  **注:** オブジェクトに他のプラットフォームのフォントが設定されている場合、その高さはカレントのプラットフォーム用に選択されたフォントに適するよう自動で調整されます。オブジェクトを描画する際この点を考慮に入れるべきです。

## スタイルシートを作成する

4D ツールボックスのスタイルシートエディターを使用して、スタイルシートを作成することができます。

スタイルシートを作成するには:

1. **デザイン**メニューから**ツールボックス＞スタイルシート**コマンドを選択します。
   ツールボックスのスタイルシートページが表示されます:
   
   
   プロパティリストのスタイルシートドロップダウンリストの横にある[...]ボタンを押して、フォームエディターからこのウインドウを開くこともできます:
   
   
   エリアの表示順は、4D が現在動作しているプラットフォームによって変わります。例えば、4D がMac OS X 上で稼働している場合は、Mac OS X エリア (ならびに関連するサンプルテキスト) がページの一番上に表示されます。
   デフォルトでは"Default"スタイルシートが作成されています。このスタイルシートは変更可能です。しかし、フォームエディターでオブジェクトを作成すると、このスタイルシートがデフォルトとして使用される点に留意してください。
2. エディターの追加ボタンをクリックします。
   または
   スタイルシートリストエリアにおいてのコンテキストメニューから**追加**コマンドを選択します。
   リスト上に新規スタイルシート (デフォルトで"スタイルシートX") が作成されます。
3. スタイルシート名をクリックすると編集モードになります。スタイルシートの名前を入力します。
   名前を確定すると、リスト中のスタイルシートが文字順で並び替えられます。
4. スタイルシート設定エリアにおいて、データベースの各オペレーティングシステムごとに目的のフォント、フォントサイズ、書体のオプションを選択します。
   変更内容は自動的に保存されます。"サンプルテキスト"エリアには現在のプラットフォーム用の変更が反映されます。
   
   

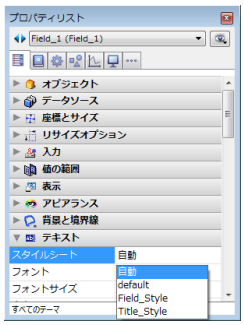
必要であれば、既存のスタイルシートを複製して、新規スタイルシートを作成することができます。この方法により、複製したスタイルシートと新規スタイルシート間に共通する項目を再度設定する必要がなくなります。
これを行うには、既存するスタイルシート上でコンテキストメニューを表示し、**複製**コマンドを選択します。既存のスタイルシートを選択して、ページのオプションメニューから**複製**を選択することもできます。

## スタイルシートを適用する

スタイルシートを使用すれば、属性を個別に設定する必要がなくなります。

作成したスタイルシートはフォームエディターのプロパティリストから使用することができます。スタイルシートはテキストを含むオブジェクトであれば適用することができます: ボタン, フィールドおよび変数オブジェクト, タブコントロール, リストボックス, 等。

オブジェクトにスタイルシートを適用するには、オブジェクトを選択し、プロパティリストのテキストテーマの、**スタイルシート**プルダウンリストからスタイルシートを選択します:

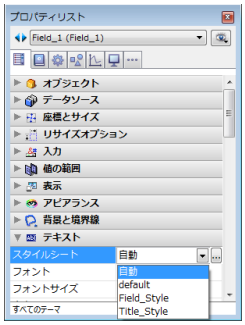



スタイルシートを選択すると、そのラベルやオブジェクトのフォント、フォントサイズ、スタイル属性が設定されます。

**注:** マルチプラットフォーム環境においては、プラットフォームによって異なるフォントが設定できるため、オブジェクトの高さがプラットフォームごとに変わるかもしれません。これはプラットフォームごとにフォントの高さが自動で設定されるためです。

## 自動スタイルシート

プロパティリストでは特別なスタイルシート"**自動**"が選択できます。




**注**: このスタイルシートはプロパティリストからのみ利用可能です。ツールボックスのスタイルシートページには表示されません。

他のスタイルシートと異なり、" 自動 " スタイルシートには属性が事前には設定されていません。オブジェクトに使用される**フォント**や**フォントサイズ**は以下のシステムパラメーターに基づいて動的に決定されます:

- プラットフォーム
- システム言語
- フォームオブジェクトのタイプ

この自動機能はデザインモードとアプリケーションモードでフォームが実行される都度適用されます。このスタイルシートを利用すればフォームオブジェクトのタイトルが現在のシステムパラメーターに準拠して表示されることを期待できます。そのサイズは条件に応じて変更されます。

"自動" スタイルシートはフォントとフォントサイズを管理します。フォームエディターで 2 つのいずれかを変更すると、そのオブジェクトの自動スタイルシートは動的に適用されなくなります。他方自動設定を変更することなくカスタムスタイルプロパティ (太字 , イタリック または下線) を設定することはできます。

4D v13 より、フォームエディターに配置される新規オブジェクトにはすべてのこのスタイルシートがデフォルトで適用されます。変換されたデータベースに既に設定されているスタイルシートは変更されません。既存のスタイルシートを自動で置換したい場合、標準の検索 /置換機能を使用することができます (デザインモードの検索と置換参照)。

## フィルターとフォーマット


カスタムフィルターとフォーマット

フィルターエディター

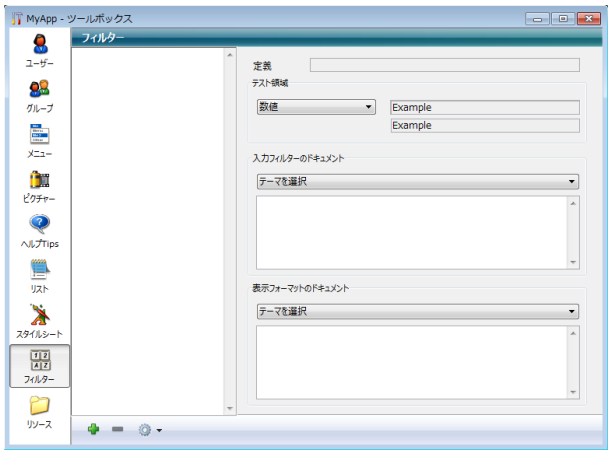
フィルターとフォーマットのコード

## カスタムフィルターとフォーマット

4Dはフォーム上でのデータ入力や表示を設定するために使用することのできる、標準の表示フォーマットや入力フィルター (データ入力の制御と補助参照) を提供します。必要であればフォーマットやフィルターをフォームのプロパティリストで直接編集できます。

フィルターエディターを使用して独自の表示フォーマットや入力フィルターを作成することもできます。作成したものは名前を使用してフォームから参照できます。カスタムフォーマットやフィルターは複数の場所で同じフォーマットやフィルターを使用する場合に便利です。同じ入力フィルターを持つ同じフィールドを複数のフォームで使用する場合、一回だけ入力フィルターを作成し、その名前をフォームエディターのフィールドプロパティで指定するだけです。さらにフォーマットやフィルターを変更すれば、その変更はフィルターやフォーマットを参照している箇所すべてに適用されます。

入力フィルターに対応する表示フォーマットを作成し、使用することができます。フォーマットや入力フィルターの作成はツールボックスの**フィルター**ページで行います:

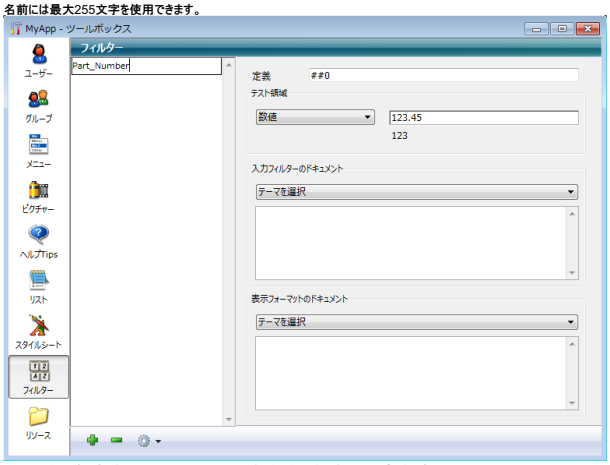


# フィルターエディター

### フィルター/フォーマットの作成と設定

通常入力フィルターと表示フォーマットはペアで作成します。

カスタムフォーマットやフィルターを作成するには:

1. 4Dのデザインメニューから**ツールボックス > フィルター/フォーマット**を選択する。
   または
   プロパティリストの入力フィルター欄にある**[...]**ボタンをクリックする。
   ツールボックスの**フィルター**ページが表示されます (最初は空)。
2. 追加ボタンをクリックするか、リストのコンテキストメニューから**追加**を選択します。
   "FilterX"という名前のフィルター/フォーマットが作成され、リストに追加されます。
3. フォーマット/フィルター名を変更します。**Alt** (Windows) または **Option** (Mac OS) キーを押しながら名前をクリックすると、いつでも編集モードに入ります。
   名前には最大255文字を使用できます。
   
   
   **注:** 目的を明確にするために"filter"や"format"といった単語を名前に含めることができます。
4. 定義エリアに表示フォーマットや入力フィルターのコードを入力します。
   定義 ##-###
   - 例えば郵便番号の表示フォーマットは以下のようになります:
     ###-####
   - 部品番号が"XA-654-1"のような形式である場合、入力フィルターを以下のように定義できます:
     !X&"A-Z"##-!0&"0-9"###-#
     そしてこれに対応する表示フォーマット"##-###-#"も作成します。

   表示フォーマットと入力フィルターの作成に関する詳細はフィルターとフォーマットのコードを参照してください。
   **注:** ウィンドウ下部にあるドキュメントエリアをダブルクリックして、エリアに入力することができます。ドキュメントエリアに関する詳細は後述の"ドキュメントエリアを使用する"を参照してください。
5. さらにフォーマットやフィルターを作成するには追加ボタンをクリックするか、エリアのコンテキストメニューから**追加**を選択します。
   または
   既存のフォーマットやフィルターを元に新規項目を作成するには、エリアのコンテキストメニューまたはページのオプションメニューから**複製**を選択します。

フィルターやフォーマットを選択して、名前やコードを変更できます。削除するには、対象を選択し、削除ボタンをクリックするかコンテキストメニューから**削除**を選択します。

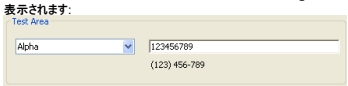
ツールボックスのフィルターエディターで作成したカスタムフォーマットやフィルターには自動で文字や数字フォーマットリストの先頭に追加されます。そのとき、名前の先頭には縦棒 (|) が追加されます (データ入力の制御と補助とGET LIST ITEM PARAMETER参照)。ビルトインフォーマットと同じようにカスタムフォーマットを選択できます。

### テスト領域を使用する

テスト領域ではカスタムフィルターやフォーマットを検証できます。

まずフィルターやフォーマットを適用するデータの型 (文字、数値、日付、時間) をメニューから指定します。

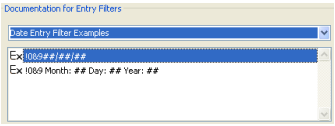
次に実際に値を入力できます:

- 入力フィルターの場合、データ入力中にデータの検証が行われます。
- 表示フォーマットの場合、データ入力後に**Enter**または**Carriage return**キーを押します。表示フォーマットを適用した結果は入力エリアの下に表示されます:
  
  

### ドキュメントエリアを使用する

フィルターエディターの"入力フィルターのドキュメント"と"表示フォーマットのドキュメント"エリアにはメニューと表示エリアがあります。

メニューからテーマを選択すると、そのテーマに関連する情報や例題が表示エリアに表示されます:




例題をダブルクリックすると定義エリアに直接入力することができます。

**入力フィルター:**

- *プレースホルダ変更*
- *イニシエータ*
- *入力フィルタ*
- *引数*
- *プレースホルダ*
- *括弧*
- *日付入力フィルタ例*
- *時間入力フィルタ例*
- *電話番号入力フィルタ例*
- *数値入力フィルタ例*
- *その他の入力フィルタ例*

**表示フォーマット:**

- *プレースホルダ*
- *括弧*
- *日付表示フォーマット例*
- *時間表示フォーマット例*
- *電話番号表示フォーマット例*
- *数値表示フォーマット例*

## フィルターとフォーマットのコード

この節では文字型入力フィルターや表示フォーマットで使用されるコード文字について説明しています。数値型の表示フォーマットに使用されるコード文字については表示フォーマットを参照してください。

**入力フィルター**コードは3つのパーツで構成されます:

*イニシエーター "引数" プレースホルダー*

- イニシエーターは、そのあとに続く引数をデータ入力中のフィルターとして使用するよう、4Dに指示します。
- 引数では許可される文字を指定します。
- プレースホルダーは文字を入力できる場所に表示される代替え文字です。

例えば以下の入力フィルターは"a"、"b"、"c"、または"g"のうち2文字だけ入力を許可します:

&"a;b;c;g"##

この例題ではANDマーク (&) がイニシエーターで、"a;b;c;g"が引数、シャープ (#) がプレースホルダーです。結果"ag"、"gc"、"ba"、"ab"、"aa"、"ac"、その他4文字の中から2文字を組み合わせたテキストを入力できます。

入力フィルターは組み合わせることができます。以下の入力フィルターは"a"、"b"、"c"、または"g"のうち2文字を受け付け、次に1、3、または8を1文字受け付けます:

&"a;b;c;g"#,#&"1;3;8"#

ユーザーはまず4文字の中から2文字を入力し、次に3つの数字から1つを入力しなければなりません。

**表示フォーマット**はプレースホルダーと標準文字の組み合わせです。

### イニシエーター

イニシエーターは2つあります。ひとつはAND (&) でもうひとつはチルダ (~) です。これらの文字は後に続く引数とプレースホルダーを入力フィルターとして使用するよう4Dに指示します。

さらにチルダ (~) は入力された文字を大文字に変換するよう4Dに指示します。小文字の入力が拒否されるのではなく、入力された小文字が大文字に変換されます。

以下2つの入力フィルターは同じ意味です:

&"P"#
~"p"#

異なる点は、AND (&) で始まるフィルターは小文字 "p" の入力を受け付けません。他方チルダ (~) で開始されたフィルターは小文字 "p" を受け入れ、大文字に変換します。

以下のケースはまったくの等価です:

&"1;5;8"#
~"1;5;8"#

### 引数

イニシエーターの後に記述されるフィルター引数は、プレースホルダーに入力できる文字種を限定します。フィルター引数を記述するには、入力を許可する文字をダブルクォーテーションで囲みます。

引数には小文字、大文字、数字、記号 (!@#$%^&*()}{[]":;?><,./~) を指定できます。小文字を指定した場合、小文字のみを入力できます。大文字を指定した場合、大文字のみを入力できます。

- 引数に一文字 (文字または数字) のみを指定することができます。例えば"j"、"J"、"6"等
- 引数に複数の文字セットを指定することもできます。文字と文字はセミコロンで区切ります。例えば"a;r;t"や"1;5"。
- 引数に文字範囲を指定することもできます。範囲を指定するには最初の文字、ハイフン、そして最後の文字を記述します。例えば"a-c"や"1-5"のようになります。"a-c"は"a;b;c"と、"1-5"は"1;2;3;4;5"と等価です。
- 引数は上記の組み合わせを指定することもできます。例えば"a;m-z;3;5-9"です。

以下は引数として特別な意味を持つ文字です。これらの文字はクォーテーションマークなしで指定します:

| 文字 | 意味 | 等価な引数 |
|---|---|---|
| 9 | 数字を許可 | "0-9" |
| a | 小文字と大文字を許可 | "a-z;A-Z" |
| A | 大文字を許可 | "A-Z" |
| @ | 文字と数字を許可 | "a-z;A-Z;0-9" |

以下のフィルターはすべて等価です:

&9#
&"0-9"#
&"1;2;3;4;5;6;7;8;9;0"#

以下のフィルターは等価です:

&a#
&"a-z;A-Z"#

以下のフィルターは等価です:

&A#
&"A-Z"#

### プレースホルダー

文字フィルターやフォーマットではシャープ (#) のみをプレースホルダーとして使用できます (数値フィルターとフォーマットでは他の文字も利用できます)。1文字につきひとつのシャープを使用します。

例えば以下の入力フィルターでは4文字を入力できます:

&a####

以下の入力フィルターでは大文字を3つ、次に数字を2つ入力できます:

&A###&9##

プレースホルダーを指定しない場合、フィルターは文字数を制限しません。以下の入力フィルターは数字の入力のみを許可しますが、桁数の制限はありません:

&9

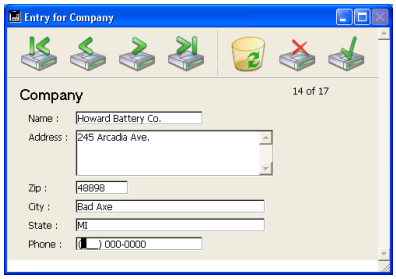
**注:** 文字フィールドではストラクチャーエディターで最大文字数を設定できます (4Dのフィールドの型参照)。

#### 表示される文字

データを入力するために、入力フィルターが設定されたフィールドにフォーカスが入ると、4Dはプレースホルダーの位置に下線 (_) を表示します。ユーザーが有効な文字をタイプすると、ハイライトされた下線部分にタイプされた文字が入力されます。
入力フィルターを感嘆符 (!) と代替え文字で開始することにより、下線を指定した文字に置き換えることができます。

例えば"XXXX"と表示し、ユーザーが許可された文字のうち例えば"AA"とだけ入力したとすると、レコードが保存される際そのデータは"AAXX"となります。

以下の図は選択されたフィールドで下線と0が表示されています。




### スキップ文字

すべての文字をスキップ文字として使用できます。スキップ文字はデータ入力中に表示されますが、挿入ポイントはスキップされ、その場所にデータは入力されません。

スキップ文字として使用したい文字はプレースホルダーの前、後、あるいは中間におかれます。データ入力中、スキップ文字は表示されます。

電話番号入力フィルター (&9(###) ###-####) では括弧、スペース、そしてハイフンがスキップ文字として使用されています。スキップ文字の前に数字を入力すると、挿入ポイントはスキップ文字の次のプレースホルダーに移動します。

# リソースエクスプローラー


概要

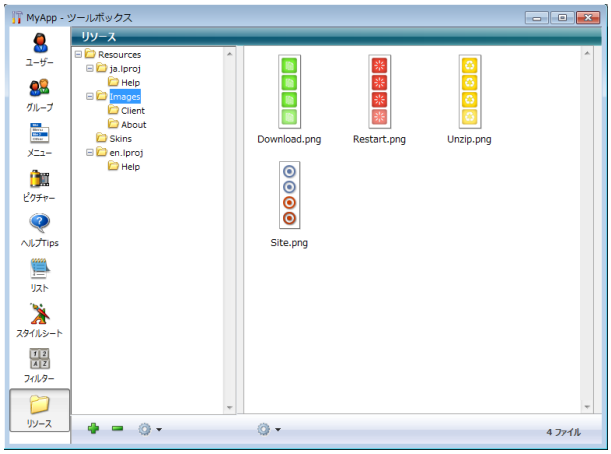
リソースエクスプローラーを使用する

クライアント/サーバーモードでのリソース同期

## 概要

4Dではデータベースストラクチャーファイル (.4dbまたは.4dc、4Dファイルの説明参照) と同階層のResourcesフォルダーに、アプリケーションで使用するリソースを格納します。このフォルダーにはデータベースやコンポーネントで使用するすべての"リソース" (テキストの翻訳、ピクチャーなどアプリケーションインターフェースのカスタマイズに使用するファイル) を格納できます。

クライアント/サーバーアーキテクチャーにおいてResourcesフォルダーの内容を管理するためのツール、リソースエクスプローラーが提供されます。このツールは4Dのツールボックスにあります:




リソースエクスプローラーにはカレントデータベースのResourcesフォルダーの内容が階層リストで表示されます。またフォルダーの内容の管理を容易にするための追加の機能 (項目の追加や削除、プレビューなど) が複数用意されています。

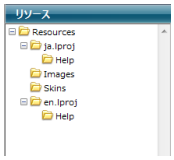
リソースエクスプローラーはローカルモードおよびリモートモードいずれでも利用できますが、リモートモードにおいて最も利用価値があります。この場合、リソースエクスプローラーを使用して4D Serverに接続しているすべてのリモートマシンのResourcesフォルダーの同期を制御できます。リソースエクスプローラーを使用すればクライアント/サーバー環境でのリソース共有を管理できます。通知メカニズムを使用してResourcesフォルダーの内容が変更された際、クライアントマシンにそれを知らせることができます。そして各クライアントマシンはその内容を同期できます。

## リソースエクスプローラーを使用する

リソースエクスプローラーを使用してカレントデータベースのResourcesフォルダーの内容を確認したり、修正したりできます。リソース要素の追加や削除、検索、そして表示などが可能です。
リソースエクスプローラーにはフォルダーリストとプレビュー、2つのエリアがあり、各エリアの下には制御ボタンがあります。

### フォルダーリスト

フォルダーリストには、データベースのResourcesフォルダー内にあるフォルダーの階層が表示されます。




フォルダーを選択、展開、折りたたみすることができます。フォルダー名をクリックすると、プレビューエリア (リソースエクスプローラーの右側) にはそのフォルダーに含まれるすべてのファイルが表示されます。ウィンドウの下部にはコントロールエリアがあり、ボタンやアクションメニューが配置されています。
フォルダーリストの下にはコントロールエリアがあり、や ボタン、およびアクションメニューがあります。ボタンは選択したフォルダーの中、またはフォルダーが選択されていない場合トップレベルにフォルダーを作成します。ボタンは選択したフォルダーとその内容を削除します。

フォルダーリストのコンテキストメニューを使用することもできます。このメニューのコマンドはアクションメニューのそれと同じです。

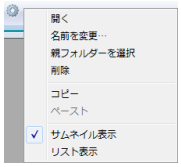
アクションメニューのコマンドは 以下の通りです:

- **新規フォルダー…**: 選択したフォルダーまたはフォルダーが選択されていない場合トップレベルに、フォルダーを作成します。このコマンドを選択すると、作成するフォルダーの名前を指定するためのダイアログボックスが表示されます。フォルダーはディスク上に物理的に作成されるため、システムが許可しない文字を含めることはできません。
- **名前を変更…**: 選択したフォルダーの名称を変更するためのダイアログボックスが表示されます。
  **注:** Resources フォルダーの名称を変更することはできません。
- **ディスク上に表示** (ローカルモードでのみ有効): 選択したフォルダーをエクスプローラー (Windows) またはFinder (Mac OS) で表示します。
- **削除**: 選択したフォルダーとその中身を削除します。
  **注:** Resources フォルダーは削除できません。
- **すべてを拡げる/すべてを折りたたむ**: リスト中のフォルダー階層をすべて展開/折りたたみます。
- **クライアントに通知** (リモートモードでのみ有効): Resourcesフォルダーの中身が更新されたことを他のクライアントマシンに通知するために使用します。通知を受け取ったクライアントは一般設定あるいはローカル設定に基づき、Resourcesフォルダーの同期を即座にあるいは後で行うことができます (クライアントマシンの更新参照)。
  Resourcesフォルダーに変更を加えたらこのコマンドを使用して、他のクライアントに同期を要求します。
- **ローカルリソースを更新** (リモートモードでのみ有効): Resourcesフォルダーの内容をサーバーのそれと強制的に同期します。
  フォルダーの同期が手動に設定されていて、サーバーマシンのResourcesフォルダーの内容が変更された旨の通知を受け取ったら (クライアントマシンへの通知参照) このコマンドを使用して同期を実行できます。
  **Shift**キーを押しながら**ローカルリソースを更新**コマンドを選択すると、サーバー上のResourcesフォルダー全体をダウンロードし、ロカールのそれと入れ替えることができます。

### プレビューエリア

リソースエクスプローラーの右側はプレビューエリアであり、選択したフォルダー (およびそのサブフォルダー) に含まれるファイルを表示します。このエリアはフォルダーリストでフォルダーを選択するたびに更新されます。

プレビューエリアの下にはアクションメニューがあります:




- **開く**: 選択されたファイルをデフォルトのエディターで開きます。これはファイルをダブルクリックするのと同じです。
- **名前を変更…**: 選択したファイルの名前を変更するためのダイアログボックスを開きます。
- **親フォルダーを選択**: フォルダーリスト中で、選択したファイルの親フォルダーを選択します。このコマンドを使用すると、Resourcesフォルダー階層のなかでのファイルの位置を知ることができます。プレビューエリアには選択されたフォルダーだけでなく、そのサブフォルダーの内容も表示される点に留意してください。
- **削除**: 選択されたファイルを削除します。
- **コピー**: 選択されたファイルをクリップボードにコピーします。
- **ペースト**: クリップボードにピクチャーまたはファイルのパス名が含まれる場合、その内容を選択されたフォルダーにペーストします。4Dは対応するタイプのピクチャーファイルを作成します。作成するファイルの名前を指定するためのダイアログボックスが表示されます。

#### サムネール表示/リスト表示

これらのコマンドはプレビューエリアの表示モードを切り替えるために使用され、相互排他的に選択されます。

- **サムネール表示** モードでは、選択されたフォルダーのファイルはサムネールとして表示されます。ピクチャータイプのファイルは (ピクチャーフォーマットを4Dが認識できれば) 内容がプレビューされます。他のタイプのファイルはシステムアイコンで表示されます。

このプレビューモードでは、コンテキストメニューを使用してファイル編集コマンドにアクセスできます。
各ピクチャーに関連付けられるヘルプTipからは追加の情報が提供されます: パス名 (相対)、名称、タイプ、更新日時、サイズ。

- **リスト表示** モードでは、選択されたフォルダーのファイルは表形式で表示されます。各ファイルごとに名前、更新日時、サイズ、そしてタイプが表

示されます:

| 名称 | 更新 | サイズ | タイプ |
|---|---|---|---|
| Download.png | 2012/02/02 1:44:46 | 11 KB | PNG ファイル |
| Restart.png | 2012/02/02 1:44:46 | 12 KB | PNG ファイル |
| Unzip.png | 2012/02/02 1:44:46 | 14 KB | PNG ファイル |
| Site.png | 2012/02/02 1:44:44 | 9 KB | PNG ファイル |

列のヘッダーエリアをクリックして、表をソートできます。

**ドラッグ＆ドロップの管理**

リソースエクスプローラーに対してドラッグ＆ドロップ操作を行い、Resourcesフォルダーやサブフォルダーに項目を追加できます:

- フォルダーリスト中でドラッグ＆ドロップを行い、Resourcesフォルダー内のフォルダー階層を変更できます。
- OSのデスクトップからドラッグ＆ドロップでResourcesフォルダーにファイルを追加できます。4Dは自動でドロップされたファイルのコピーをResourcesフォルダー内に作成します。フォルダーリストまたはプレビューエリアにドロップできます。
- ResourcesフォルダーにOSから、または他の4DアプリケーションのResourcesフォルダーから、フォルダーをドロップすることができます。他の4Dアプリケーションからフォルダーがドロップされた場合、フォルダーは移動されます。それ以外の場合ファイルと同様にコピーを作成します。フォルダーはフォルダーリストまたはプレビューエリアにドロップできます。後者の場合、プレビューエリアに表示されている階層レベルに対応する場所にフォルダーが作成されます。
- Resourcesフォルダーからフォームエディターやメソッドエディターにピクチャーをドラッグ＆ドロップできます。ピクチャーは参照として挿入されます。

## クライアント/サーバーモードでのリソース同期

リソースエクスプローラーはクライアント/サーバー環境での共同開発を容易にします。すべての接続したマシンでResourcesフォルダーの内容をリアルタイムに同期させることができます。

Resourcesの同期を行うステップは以下の通りです:

1. クライアントマシン上でResourcesフォルダーの内容を更新した場合、サーバー上のResourcesフォルダーを更新する。
2. すべての接続ユーザーに通知する。
3. 接続さr他クライアントマシン上でリアルタイムに更新される。

**注**: 接続していないユーザーは、接続したときに自動でResourcesフォルダーの同期が行われます。

### サーバーの更新

このモードは以下のように動作します。4DクライアントのResourcesフォルダーに対して、リソースエクスプローラーでローカルに行われた更新は、自動的にサーバーに転送されます。

### クライアントマシンへの通知

接続中のクライアントマシンはResourcesフォルダーの内容が更新されると通知を受けとります:

- 4Dクライアントから行われた最後の更新から2分経過した後、サーバーが自動で行う (この遅延により、多数のファイルがコピーされるような場合にそれだけ通知が発生することを避けるようになっています)。
- 更新を行ったクライアントマシン上でリソースエクスプローラーの**クライアントに通知**コマンドを使用する (リソースエクスプローラーを使用する参照)。
- NOTIFY RESOURCES FOLDER MODIFICATIONコマンドを使用する。このコマンドはストアドプロシージャーを使用してサーバーマシン上でResourcesフォルダーが更新されたような場合に便利です。

クライアント側では、通知を受け取った際に各マシンの設定に基づいてアクションが実行されます (次節参照)。

### クライアントマシンの更新

サーバー上のResourcesフォルダーの内容が更新された旨の通知を受け取ったら、各クライアントマシンはその内容を同期できます。この同期はアプリケーションのグローバル設定または個々のクライアントの設定に基づき、自動または手動で行えます。

この設定はデータベース設定を使用してグローバルに、またはSET DATABASE PARAMETERコマンドを使用して個々に設定できます。

更新モードの設定はデータベース設定のクライアント-サーバー/ネットワークオプションページの**セッション中に"Resources"フォルダーを更新**を使用して行います。

3つのオプションが用意されています:

- **しない**: サーバーからの通知は無視され、ローカルのResourcesフォルダーは次回の接続時、またはリソースエクスプローラーで**ローカルリソースを更新**コマンドを使用したときに更新されます。
- **常に**: ローカルResourcesフォルダーの同期はサーバーから通知を受け取ったときに自動で行われます。
- **その都度指定**: サーバーから通史を受け取った際に、クライアントマシン上でダイアログボックスが表示されます。ユーザーはローカルResourcesフォルダーを同期するかどうか決定できます。

**注:** サーバー上の環境設定で設定が実行された場合、その設定はすべてのクライアントマシンに適用されます。クライアントマシン上での設定は、そのマシンにのみ適用されます。

## レコードの管理


テーブルやフォームをブラウズする

レコードの表示と選択

レコードを編集する

レコードを印刷する

# テーブルやフォームをブラウズする

データベースにおいて情報はテーブルに格納されます。各テーブルは特定の情報を扱います。例えば"sales contacts"データベースには個人情報を格納するテーブルと会社関連の情報を格納するテーブルがあるでしょう。

データを入力したり表示したりするためにはフォームを使用します。データベースの各テーブルはカレントの入力フォームと出力フォームを持ちます。入力フォームは一度にひとつのレコードを表示します。入力フォームを使用して1レコードに対しデータを入力したり、表示、あるいは更新を行ったりします。出力フォームは複数のレコードを一覧形式で表示します。出力フォームでは複数レコードのなかから目的のレコードを探したり、レコードを選択したり、一覧の印刷を行ったりできます。出力フォーム中で直接レコードを作成したり変更したりすることもできます。

レコード表示ウィンドウに表示するテーブルを別のテーブルに切り替えたり、入出力フォームを変更したりできます。

アプリケーション環境では通常このような変更はカスタムインターフェースを使用して行います。テーブルやフォームの切り替えは4Dランゲージコマンドを使用して行います。

デザインモードでは特別なコマンドを利用して、いつでもテーブルやフォームを切り替えられます。

各テーブルはカレント入出力フォームを持ちます。これらのデフォルトフォームはデータベースの各テーブルごと、エクスプローラーを使用して指定します。セッション中は他のフォームを指定するまで、自動でこれらのフォームが使用されます。

データを表示するテーブルを切り替えるには、4Dツールバーの**テーブル**ボタンを使用します。表示するテーブルやフォームを選択するには、**テーブルリスト**ウィンドウを使用します。このウィンドウを使用すれば素早くテーブルやフォームを切り替えることができます。入出力フォームを使用している最中でも、テーブルやフォームを切り替えることが可能です。選択は即座に適用されます。

### レコードの表示

レコード表示ウィンドウを最前面にするには:

1. 4Dツールバーの**テーブル**ボタンをクリックします。

または
**レコード**メニューからカレントテーブル表示 (テーブル名) を選択します。
カレントテーブルのデータがカレント出力フォームを使用して表示されます。

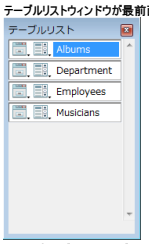
レコードを表示するテーブルを変更するには:

1. **テーブル**ボタンに関連付けられているリストからテーブルを選択します。
または
**レコード**メニューの**最後に使用したテーブル**サブメニューからテーブルを選択します。
リストにはセッション中に使用されたテーブルの名前が表示されます:

選択したテーブルのデータが、そのテーブルのカレント出力フォームに表示されます。

### テーブルリストウィンドウ

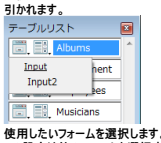
テーブルリストウィンドウを使用してテーブルやフォームを選択するには:

1. **レコード**メニューから**テーブルリスト**コマンドを選択します。
テーブルリストウィンドウが最前面になります。


必要であればスクロールバーやウィンドウのリサイズを使用して、目的のテーブルを表示します。
2. ウィンドウ中でテーブル名をクリックします。
4Dは出力フォームを使用してこのテーブルのカレントセレクションを表示します。

ウィンドウ中の各テーブルの左には入力フォームと出力フォームを表す2つのアイコンが表示されます。左のアイコンをクリックすると入力フォームのリストが表示され、右側のアイコンをクリックすると出力フォームのリストが表示されます。

カレントフォームを変更するには:

1. カレントフォームを変更したい入力または出力フォームのアイコンをクリックします。
選択されたテーブルの入力または出力フォームのリストがドロップダウンメニューが表示されます。現在使用されているフォームの名前には下線が引かれます。


2. 使用したいフォームを選択します。
この設定は他のフォームを選択するか、データベースを終了するまで有効です。

**注:** デフォルトの入力および出力フォームはエクスプローラーの[フォームページ]で指定します。詳細は[詳細/一覧フォームを指定する]を参照してください。

# レコードの表示と選択

### カレントセレクション

データを管理する際、作業対象のレコードグループを選択します。このレコードグループを4Dではカレントセレクションと呼びます。カレントセレクションにはレコードが全く含まれない、あるいは1件のレコード、複数のレコード、またはテーブルのレコード全部が含まれる場合があります。各テーブルには各プロセスごとにカレントセレクションがあります。

4Dにおいてカレントセレクションは重要なコンセプトです。通常データ管理処理はカレントセレクションのレコードに対して行われます。例えば:

- レコードの並び替え
- 個々のレコードの表示と更新
- レコードグループへの更新
- レポートの印刷
- ラベルの生成
- データのグラフ化
- レコードの書き出し

つまりカレントセレクションの作成は、あらゆるデータ管理処理の最初のステップです。

レコードのカレントセレクションはテーブルから抽出されたレコードのセットです。例えば"Company"データベースに従業員を管理するための"Employee"テーブルがあるとき、この会社のエンジニアを検索することができます。
検索を実行する前、そのプロセスの"Employees"テーブルのカレントセレクションは会社のすべての従業員 (営業、人事、エンジニア等) レコードを含んでいるかもしれません。検索が実行された後は、カレントセレクションにはエンジニアだけが含まれます。レコードリストを印刷すると、そのリストにはカレントセレクションのレコード (この場合エンジニア) だけが含まれます。従業員の給与をグラフ化すると、すべてのエンジニアの給与がグラフ化されます。

カレントセレクションは、それを変更する処理を行うまでは変更されません。カレントセレクションは以下のような場合に変更されます:

- すべてのレコードをカレントセレクションにした
- ハイライトしたレコードをカレントセレクションにした
- レコードを検索した

出力フォームのウィンドウのタイトルバーにはテーブルの総レコード数とカレントセレクションのレコード数が表示されます。

**4D Server**: 4Dクライアントの場合、カレントセレクションのレコード数だけが表示されます。

入力フォームにはカレントセレクションのレコード総数と、表示中のレコードがカレントセレクション中何番目のレコードかが、ナビゲーションボタンの下に表示されます (表示のための変数が配置されていれば)。

データベース中のすべてのテーブルが固有のカレントセレクションを持ちます。リレーショナルデータベースでは、1テーブルのカレントセレクションを変更すると、そのテーブルにリレートする他のテーブルのカレントセレクションも変更されることがあります。例えば[Employees]と[Departments]テーブルがリレートされているとき、[Departments]テーブルのレコードを詳細フォームで開くと、[Employees]テーブルのカレントセレクションが変更されます。つまりその部署に属する従業員が[Employees]テーブルのカレントセレクションになります。リレーションに関する詳細はリレーションのタイプを参照してください。

データベースタスクを実行するためにプロセスを使用している場合、あるテーブルに対して同時に複数のカレントセレクションが存在できます。各プロセスは個々の4D環境のように振る舞い、それぞれ個別のタスクを実行できます。異なる複数の営業地域間で月間売り上げの比較を行うなど、同じデータの異なるカレントセレクションを同時に扱う必要がある場合に複数のプロセスを使用できます。プロセスに関する詳細は4Dランゲージリファレンスマニュアルのプロセスを参照してください。

### すべてのレコードを表示する

レコードを表示する際に出力フォームを使用している場合、カレントセレクションをリセットしてすべてのレコードを表示することができます。

アプリケーションモードでは"すべてを表示"標準アクション、またはALL RECORDSコマンドを使用します。

デザインモードでは**レコード**メニューの**すべてを表示**コマンドを使用できます。これを選択すると、カレントテーブルのすべてのレコードがカレントセレクションに含まれます。

**注:** 入力フォーム使用時には**すべてを表示**コマンドを選択できません。

### ハイライトしたレコードをカレントセレクションにする

出力フォームに表示されているレコードをクリックして選択し、それらを新しいカレントセレクションにすることができます。この動作はサブセットを作成すると呼ばれることがあります。

アプリケーションモードではユーザーがハイライトしたレコードをセットとセレクションテーマのコマンドで処理します。

デザイン環境では**レコード**メニューの**サブセット表示**コマンドを選択します。このコマンドを選択すると、4Dはウィンドウ中で選択 (ハイライト) されたレコードを新しいカレントセレクションとします。

レコードをハイライトする様々な方法については一覧フォームの"ハイライト"で説明しています。

## レコードを編集する

ここではデータベースを取り扱う際に実行する以下の主要な操作について説明します:

- レコードの追加
- レコードの更新
- レコードの削除

デザインモードでデータベースを作成したら、レコードの追加や更新、削除を開始することができます。これらの処理は通常アプリケーションモードのカスタムインターフェースで行います。

4Dではデザインモードでもレコードの追加、更新、削除を行うことができます。これにより基本的なタスクやアプリケーションデータのテストを行うことができます。

データベースにレコードを追加した後で、ほとんどの場合はデータの修正が必要になります。1件以上のレコードを修正する必要がある場合、レコードセレクションメソッドを使用してレコードを表示します (レコードの検索参照)。そして出力あるいは入力フォームを使用してレコードを更新できます。

また、テーブル内の一連のレコードに対して、まったく同じ変更が必要となる場合があります。これは"グローバル更新"と呼ばれます。4D では、各レコードを個別に変更しなくても、テーブルのカレントセレクションを自動的に更新することができます。

さらに、1 件以上のレコードの削除が必要となることもあるでしょう。あるレコードが無効であるか、不適切であることを発見した場合は、レコードを削除することができます。レコードは必要でも、レコードに格納された値が正確ではなくなった場合は、削除でなく修正を行います。入力フォームからカレントレコードを削除したり、または出力フォームを用いてカレントセレクション内の一部のレコードを削除することができます。

### レコードを追加する

入力フォームまたは出力フォームを使用して、レコードを追加することができます。

アプリケーションモードでは"サブレコード追加"標準アクション (リストにレコード追加) またはADD RECORDやCREATE RECORDコマンドを使用して、レコードを追加できます。

デザインモードでは、4Dが提供するレコード追加機能を使用して、入出力フォームでレコードを追加できます。

- 入力フォームを使用して新規レコードを追加するには:
  1. デザインモードの**レコード**メニューから**新規レコード**コマンドを選択します。
     入力および出力フォームいずれかを使用しているときに新規レコードコマンドを選択できます。
     4Dは空の入力フォームを表示し、フォーム中の最初の入力可オブジェクトに入力カーソルを置きます。
  2. 入力可能オブジェクトにデータを入力したら**Tab**または**Carriage return**を押して次のエリアに移動します。
     すべてのデータを入力したら、レコードを保存できます。
  3. テンキーの**Enter**キーを押すか、入力フォームの**OK**ボタンをクリックするとレコードが受け入れられます。
     **注**: キーボードの割り当てはデータベース設定のOBJECT Get enabledで変更できます。
     レコードを受け入れると、4Dは (必要であれば値を検証し) データをデータベースに格納します。そして空の入力フォームが再び表示されます。
  4. さらにレコードを追加したい場合、データ入力とレコード受け入れ操作を繰り返します。
     または
     データの入力を終了する場合、フォーム上の**キャンセル**ボタンをクリックするか**Esc**キーを押します。
     **注**: キーボードの割り当てはデータベース設定のOBJECT Get enabledで変更できます。
     すると出力フォームが表示されます。

出力フォーム上で直接レコードを追加することもできます。この場合変数や他のテーブルのフィールド、サブフォームには入力できない点に留意してください。

- 出力フォームを使用して新規レコードを追加するには:
  1. デザインモードで**レコード**メニューから**新規レコードをリストに追加**コマンドを選択します。
     出力フォームに表示されている最後のレコードの下の先頭フィールドに入力カーソルが表示されます。
  2. フィールドにデータを入力し、**Tab**や**Carriage return**キーで次のフィールドに移動します。
     **注: Shift+Tab**や**Shift+Carriage return**を使用するとレコードのフィールドを逆方向に移動します。
     種るよくフォームのフィールドに割り当てられているデータ検証制御はそのフィールドを選択したときに使用されます。例えば選択リストが割り当てられている場合、そのフィールドに入力カーソルが入ると、選択リストが表示されます。
  3. テンキーの**Enter**を押すとレコードが保存され、新しい空のレコードが作成されます。
     または
     他のレコード上のフィールドをクリックします。
     4Dは追加されたレコードの入力を受け入れます。

### レコードを更新する

情報を更新する必要がある場合、または元の情報が正しくないことが判明した場合、レコードを修正する必要があります。レコードグループを更新するにはまず更新対象のレコードをカレントセレクションにしなければなりません。そのためにレコードを検索するか、出力フォーム上で対象レコードをハイライトし、サブセットとして選択します。

入力および出力フォームを使用してレコードを更新できます。出力フォームでは複数のレコードを一度に見ることができるので、一連のレコードを更新するような場合には便利です。しかし通常出力フォームはすべてのフィールドを含んでいません。また入力フォームに細かく指定されているような入力制御がないかもしれません。
他のプロセスまたは他のユーザーがレコードを更新中の場合、その編集中のレコードはロックされているといわれます。ロックされたレコードを見ることはできますが、編集することはできません。ロックされたレコードを開くとフィールドデータを見ることはできます。しかしデータを入力することはできません。

アプリケーションモードでは"サブレコード編集"標準アクション (リストフォーム中でレコードを更新) またはMODIFY RECORDコマンドでレコードを編集できます。

デザインモードでは4Dが提供する複数の機能を使用してレコードを更新できます:

- 入力フォームを使用してレコードを更新するには:
  1. 出力フォーム中でレコードをハイライトし、**レコード**メニューから**レコード修正**を選択します。
     または
     出力フォーム中で対象レコードをダブルクリックします。
     4Dはデフォルトの入力フォームを使用してレコードを開きます。
  2. 編集するフィールドをクリックして挿入カーソルを入れ、データを更新します。
  3. **OK**ボタンをクリックするかテンキーの**Enter**キーを押して更新したレコードを受け入れ、出力フォームに戻ります。
     または
     ナビゲーションボタン (**前レコード**, **先頭レコード**, **次レコード**, **最終レコード**標準アクション) をクリックしてレコードを受け入れ、カレントセレクションの他のレコードに移動します。
     標準アクションナビゲーションボタンをクリックすると、カレントセレクション中でカレントレコードを移動します。
     編集をキャンセルして出力フォームに戻るには**キャンセル**ボタンをクリックするか、**Esc**キーを押します。

出力フォームに表示されているデータを直接編集することも可能です。この場合変数、他のテーブルのフィールド、サブフォームにはデータ入力できない点に留意してください。
アプリケーションモードではリストフォームでのレコード更新可/不可をコントロールすることができます。

- 出力フォームを使用してレコードを更新するには:
  1. レコードを選択し、編集するフィールド上をクリックします。
     出力フォームのフィールドが編集モードになります。
     **注:** デザインモードでは**レコード**メニューから**レコード修正**コマンドを選択すると、入力フォームでレコードを修正できます。
  2. 新しいデータを入力し**Tab**または**Carriage return**キーを押します。
     4Dは変更を保存し、次のフィールドを選択します。
  3. 必要なだけ編集作業を繰り返します。
  4. フォーム上の他のレコードのフィールド上をクリックすると、編集モードを終了します。

### グローバル更新

一連のレコードに対して一定の変更を行いたい場合は、グローバル更新を行うことができます。グローバル更新を行い、面倒で時間がかかるレコードグループへの変更を自動化できます。例えば次のような処理を行いたい場合グローバル更新を実行します:

- 在庫テーブルのすべての価格を一定の割合で変更する。
- 数値や文字タイプのフィールドをフォーマットする。

グローバル更新は、レコードのカレントセレクションに対してフォーミュラを"適用"することによって行われます。つまりカレントセレクションの各レコードをフォーミュラを用いて変更します。

いくつかのフォーミュラの例題とそのフォーミュラが実行する機能の説明を次に示します:

- 次のフォーミュラでは給与フィールドに1.05 を乗じています。たとえば昇給が実施される際にこのフォーミュラを使用することができます:
  ```
  [Emp]Salary :=[Emp]Salary *1.05
  ```
- 次のフォーミュラでは、4D 関数を用いて部品IDフィールドのデータを大文字に変換しています。これにより部品IDデータのアルファベット部が大文字であることが保証されます:
  ```
  [Parts]PartID :=Uppercase([Parts]PartID)
  ```
- 次のフォーミュラには、ローマ字名フィールドをフォーマットするユーザー作成の関数Capitalizeが含まれています。この関数はローマ字名フィールドの最初の1 桁を大文字にし、後の文字は小文字のままにしておきます:
  ```
  [Emp]LastName:=Capitalize([Emp]LastName)
  ```

グローバル更新を実行する際にユーザー作成の関数を組み込める機能は、4D の強力な特長の1 つです。フォーミュラには4Dランゲージ関数や (開発者が使用可能に指定した) プロジェクトメソッドを指定できます。セキュリティのため、フォーミュラからのプロジェクトメソッドへのアクセスは制限されています (セキュリティページ参照)。

一行を超えるフォーミュラを書くことはできません。つまり改行を挿入して二行目を書くことはできません。しかしフォーミュラエディターで利用可能に指定されたプロジェクトメソッドを使用すれば、このメソッドに複雑な処理を記述することができます。

グローバル更新を実行するには、を使用して、カレントセレクションの各レコードに適用されるフォーミュラを記述します。これを行うには**レコード**メニューの**フォーミュラで更新...**を選択します。事前にディスク上のファイルとして保存したフォーミュラ (拡張子.4fr) をロードすることもできます。詳細はを参照してください。

アプリケーションモードではEXECUTE FORMULAコマンドを使用して直接更新フォーミュラを実行したり、EDIT FORMULAコマンドでフォーミュラエディターを表示できます。

**レコードを削除する**

あるレコードが無効になったり、不要となった場合、それを削除することができます。レコードは必要でも、レコードに格納された値が正確ではなくなった場合は、削除ではなく修正を実行します。

次の2 つの方法を使用し、レコードを削除することができます:

- レコードを1 件ずつ削除 (通常入力フォームを使用)
- 一連のレコードを一度に削除 (通常出力フォームを使用)

アプリケーションモードでは"レコード削除"や"サブレコード削除"標準コマンド、あるいはDELETE RECORDやDELETE SELECTIONコマンドを使用してレコードを削除できます。

デザインモードでは**編集**メニューのクリアコマンドや**Delete**キーでもレコードを削除できます。

**警告:** レコードの削除は恒久的であり、一度削除したレコードはバックアップからしか取り戻せません。レコードを削除するとき、4Dは本当に削除を実行してよいか確認するダイアログを表示します。

入力フォームからレコードを削除すると、削除を行う前に各レコードの内容を検証することができます。

- 入力フォームを使用してレコードを削除するには:
  1. 削除するレコードを開きます。
  2. 削除ボタンをクリックしてレコードを削除します。
     入力フォームのデザインにより、**削除**ボタンはさまざまな形式で表現されます。デフォルトで削除ボタンにはゴミ箱アイコンが表示され、"レコード削除"標準アクションが割り当てられています。(フォームに削除ボタンがない場合もあります。)
     削除を確認するダイアログが表示されます。4Dがレコードを削除した後は、それを取り消すことはできません。
  3. 削除を実行するために**OK**ボタンをクリックします。
     4Dはデータベースからカレントレコードを削除し、出力フォームに戻ります。

出力フォームを使用して複数のレコードを一回の処理で削除できます。削除するレコードは出力フォーム上でハイライトされていなければなりません。

- 出力フォームでレコードを削除するには:
  1. 削除するレコードをハイライトします。
  2. **編集**メニューから**クリア**を選択するか、**Delete**または**Backspace**キーを押します。
     削除を確認するダイアログが表示されます。4Dがレコードを削除した後は、それを取り消すことはできません。
  3. 削除を実行するために**OK**ボタンをクリックします。
     ハイライトされたレコードがデータベースから削除されます。
     **注:** テーブルのすべてのレコードを削除するには、**レコード**メニューから**すべてを表示**を選択し、**編集**メニューから**すべてを選択**を選択、そして**編集**メニューから**クリア**を選択 (または削除キーを使用) します。

**ロックされたレコードの削除**

ロックされたレコードは削除できません。レコードが別のプロセスで使用されると、そのレコードはロックされます。例えばあるプロセスで修正のためにレコードを開くと、4D は別のプロセスで修正されないように、そのレコードをロックします。

**4D Serverに関する注記**: レコードが別のユーザーにより使用されている場合もレコードがロックされます。

レコードを削除する前に、削除しようとするレコードのセレクションを作成します。セレクションにロックされたレコードが含まれている場合、その削除処理は実行されますが、ロックされたレコードは削除されず、削除後もカレントセレクション内に残されます。これらのレコードを削除するには、レコードのロックが解除される (つまり、そのレコードが使用されなくなる) まで待たなければなりません。レコードロックテーマのコマンドを使用してこのタイプのシナリオを管理できます。

**別プロセスで削除されたレコード**

別プロセスでレコードが削除されることにより、カレントセレクションが変更される場合があります。例えばデータベースの作業中に別プロセスを開始して、テーブルからあるレコードを削除したものとします。するとそのプロセスで削除されたレコードはテーブルから永久削除されます。しかし新しいレコードのセレクションが作成されるまで、現在データベースの処理中に表示しているそのレコードに対し、テーブルへのこれらの変更が反映されない可能性があります。

この点について説明するため、あるテーブルに50 件のレコードがあり、全レコードがカレントセレクションに含まれている場合について考えてみましょう。この時点では、出力フォームのタイトルバーに"50/50 (50 件中の50 件) "のレコードが選択されていると表示されます。別プロセスでこの中の1 レコードが削除されると、タイトルバーは"50/49 (49 件中の50 件) "のレコードが選択されているという表示に変わります。これではテーブル内のレコード数よりも多いレコードがカレントセレクションに存在していることになります。カレントセレクションを変更すると、タイトルバーが更新されます。

削除されたレコードを修正または削除しようとすると、ダイアログボックスが表示され、そのレコードが削除されたことを知らせます。

**4D Serverに関する注記:** 別のユーザーにより削除されたレコードも、カレントセレクションに対して同様の影響を与えます。レコードはテーブルから削除されますが、カレントセレクションからは取り除かれません。したがって、カレントセレクションには実際にテーブルに存在する数よりも多いレコードが含まれているようにみえる場合があります。

## レコードを印刷する

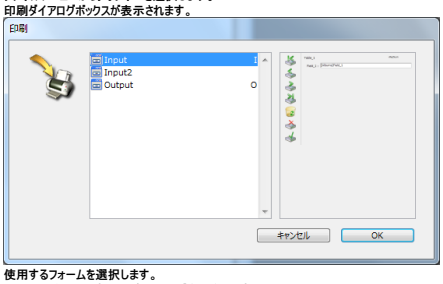
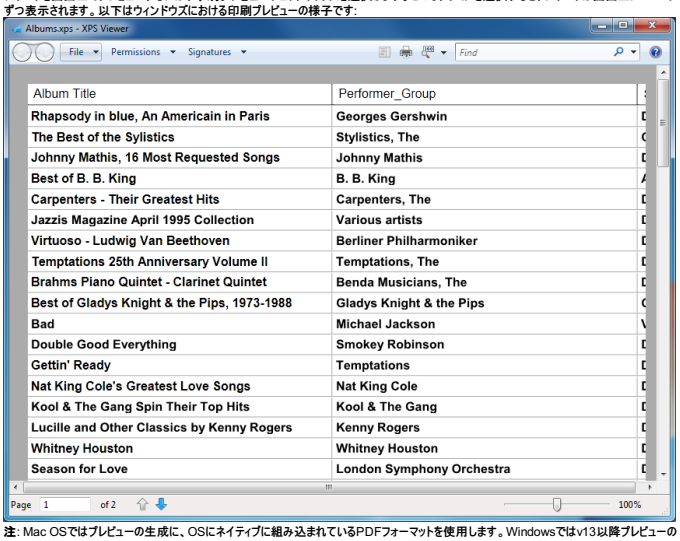
テーブルのレコードを印刷することができます。レコードはデータベースに定義されたフォームを使用して印刷されます。

アプリケーションモードではレコードの印刷をボタンやカスタムメニューを使用して管理します。レコードは専用のフォームを使用して印刷されます。印刷テーマのコマンドを使用して印刷処理をカスタマイズできます。

デザインモードでは印刷を行う際に、使用するフォームを指定します。レコードリストあるいは1つのレコードをページモードで印刷できます。

カレントセレクション内のレコードが印刷されます。印刷対象のレコードをカレントセレクションとして選択します。レコードリストを特定の順番で印刷する場合、印刷前にカレントセレクションを並び替えます。レコードをページモードで印刷する場合、そのレコードをダブルクリックして開きます。

レコードを印刷するには:

1. レコード表示ウィンドウに、印刷するレコードを表示します。
2. (オプション) レコードを並び替えます。
   このステップはブレーク処理を行うレポート印刷時には必須です。この場合レポートのブレーク数より1以上多い数の条件でソートを行わなければなりません。この点に関する詳細はレポート印刷用のフォームを参照してください。
3. **ファイル**メニューから**プリント...**を選択します。
   印刷ダイアログボックスが表示されます。
   
   
4. 使用するフォームを選択します。
   フォーム名をクリックすると、プレビューが表示されます。
5. **OK**をクリックします。
   選択されているプリンターの用紙設定ダイアログボックスが表示されます。オプションを設定し**OK**をクリックします。続いて選択されているプリンターの印刷ダイアログボックスが表示されます。
   レポートを画面上でプレビューするには、**印刷プレビュー**チェックボックスを選択します。このオプションを選択すると、レポートは画面上に1ページずつ表示されます。以下はウィンドウズにおける印刷プレビューの様子です:
   
   

| Album Title | Performer_Group |
|---|---|
| Rhapsody in blue, An Americain in Paris | Georges Gershwin |
| The Best of the Sylistics | Stylistics, The |
| Johnny Mathis, 16 Most Requested Songs | Johnny Mathis |
| Best of B. B. King | B. B. King |
| Carpenters - Their Greatest Hits | Carpenters, The |
| Jazzis Magazine April 1995 Collection | Various artists |
| Virtuoso - Ludwig Van Beethoven | Berliner Philharmoniker |
| Temptations 25th Anniversary Volume II | Temptations, The |
| Brahms Piano Quintet - Clarinet Quintet | Benda Musicians, The |
| Best of Gladys Knight & the Pips, 1973-1988 | Gladys Knight & the Pips |
| Bad | Michael Jackson |
| Double Good Everything | Smokey Robinson |
| Gettin' Ready | Temptations |
| Nat King Cole's Greatest Love Songs | Nat King Cole |
| Kool & The Gang Spin Their Top Hits | Kool & The Gang |
| Lucille and Other Classics by Kenny Rogers | Kenny Rogers |
| Whitney Houston | Whitney Houston |
| Season for Love | London Symphony Orchestra |

   **注**: Mac OSではプレビューの生成に、OSにネイティブに組み込まれているPDFフォーマットを使用します。Windowsではv13以降プレビューの生成にXPSを使用します (後述)。

レポートのページがプレビューされたら、以下の処理を行うことができます:

- **拡大** ボタンをクリックしてプレビューを拡大表示できます。拡大モードを終了数にはもう一度拡大ボタンをクリックします。
- 拡大表示中はドラッグ & ドロップを使用して表示するページ領域を変更できます。
- **次ページ**や**前ページ**ボタンを使用してページ間を移動できます。これらのボタンは移動するページがない場合選択不可になります。
- **停止** ボタンをクリックすると印刷がキャンセルされます。
- **プリント**ボタンをクリックするとプレビュー中のページが印刷されます。

**印刷プレビュー**を選択しなかった場合、レポートは即座にプリンターに送信されます。

### WindowsでのXPS印刷プレビュー

Windows において 4D v13 は XPS ドキュメントとして印刷プレビューを生成するようになりました。XPS (XML Paper Specification) フォーマットはページ記述言語に使用されるオープンな仕様で、Microsoft により開発され Ecma International により標準化されました。現在では Windows アプリケーションで広く使用され、印刷ドキュメントを正確にスクリーンに表示できます。ハードウェア設定の影響は受けません。

#### OSごとのサポート

XPS プレビュードキュメントを生成するために 4D は XPS ドライバーを呼び出します。このドライバーは Windows Vista 以降に統合されていますが、Windows XP または Windows Server 2003 でも .NET framework をインストールすることで 利用できます。

Windows のバージョンに基づき、プレビューはネイティブで利用可能なXPS ドライバーあるいは Internet Explorer を使用して開かれます。

以下の表は Windows のバージョンごとに利用可能な XPS ドライバーとビューアーのリストです:

| Windows バージョン | XPS ドライバー | XPS ビューアー |
|---|---|---|
| Windows XP (SP2) or Windows Server 2003 | なし | なし |
| Windows XP (SP2) or Windows Server 2003 + .NET 3.0 framework | あり | IEを利用 |
| Windows Vista | あり | IEを利用(*) |
| Windows 7 以降 | あり | あり(reader.exe) |

(*) 独立したビューアーもあります。

#### 4Dでの動作

4D v13 で XPS プレビューは以下のために使用されます:

- デザインモードとアプリケーションモード両方で、すべての 4D フォームとオブジェクト
- ラベルおよびクイックレポートエディター

印刷プレビュー生成が要求されると、4D は印刷を一時的な XPS ファイルに転送し、.XPS ファイルに関連付けられたアプリケーションを起動します。

**注:** 関連付けられたアプリケーションが XPS ビューアーでない場合 (例えばデフォルトブラウザーが IE でない場合)、4D は強制的に Internet Explorer を実行します。

一時的な XPS ファイルはローカルのユーザーフォルダーに保存されます:
*...:\Users\{UserName}\AppData\Local\Temp\4D\PrintPreview\FolderNumber\NumUUID*

このフォルダーがいっぱいになることを避けるために、最新 100 個のドキュメントのみが保持されます。フォルダーへの適切なアクセス権がない場合、-9799 エラーが生成されます。Windows は "Temp" フォルダーを空にすることがある点に留意してください。
ビューアーが自動でドキュメントを開きますが、まだ名前付きで保存されていません。そのために、XPS ドキュメントを格納するための UUID サブフォルダーが連番のフォルダーの下に作成されます。印刷が終了するとドキュメントに名前が付けられます。デフォルトでは印刷が起動されたときのウィンドウ名が使用されます。この名前は印刷前にSET WINDOW TITLEやSET PRINT OPTION(Spooler document name option; "xx")コマンドを使用することで変更できます。

2 つの新しいコマンドGet print previewとIs in print previewを使用してプレビューオプションをコントロールできます。

# レコードの検索


検索の基本

クエリエディター

フォーミュラによるクエリ

フォームによるクエリ

クエリ&修正

## 検索の基本

検索は、最も頻繁に用いられるデータベース操作の 1 つであり、多くの場合、作業を行おうとするレコードを選択する最も便利な方法です。

“検 索”という用語は、1 つ以上のフィールド内容に基づいて、データベースのレコードグループを探し出すことを意味します。クエリを指定することにより、検索を実行します。“クエ リ”とは、いずれのレコードを新しいカレントセレクションに含めるか (例えば、会社名が4Dのもの) を4Dに対して指示する一連の命令です。

通 常、クエリには“フィールド名”、“比較演算子”、“値”という 3 つの要素があります。フィールド名は、カレントテーブルまたはリレートテーブルのフィールドです。比較演算子は、フィールド内容と指定した値をと比較する 方法 (等しい、より大きい、より小さいなど) を4Dに対して指示します。値には、数値や文字列、または他の値を指定し、この値と各レコードとが比較されます。
例えば、給与が 3,000,000 円以上である従業員のレコードを調べたい場合は、“給与が 3000000 以上”というクエリを使用します。“給与”はフィールド、“以上”は比較演算子、“3000000”は値に相当します。
デー タベースを検索する時に、4Dはクエリ上のフィールド内容と、指定された値とを比較します。新しいカレントセレクションは、クエリで指定した条件を満たす レコードで構成されます。また、この新しいカレントセレクションには、レコードが全く含まれないか、1 レコードのみ、レコードのグループ、または
テーブルの全レコードが含まれている可能性があります。

入力フォームまたは出力フォームのいずれかを使用しながら、クエリを実行することができます。入力フォームの使用中にクエリを実行すると、新しいカレントセレクションの先頭レコードが入力フォーム上に表示されます。このレコードの参照や、修正、印刷を行うことができます。

新 しいカレントセレクションが複数のレコードで構成される場合は、レコードナビゲーションボタン (前レコード、次レコード、先頭レコード、最終レコード) を用いて、レコード間を移動することができます。レコードを修正した後にレコードナビゲーションボタンをクリックした場合、4Dにより変更内容が保存されます。リレーショナルデータベースにおいて、テーブル間にリレートが設定されている場合は、
他のテーブルのフィールドを使って検索することができます。リレートに関する詳細は、4Dデザインリファレンスマニュアルを参照してください。

出力フォームを用いてクエリを実行すると、新しいカレントセレクションが出力フォーム上に表示されます。**レコード**メニューの**すべて表示**を選択すると、カレントセレクションをカレントテーブルの全レコードにリセットすることができます。

**注:** デー タベースストラクチャー内の任意のフィールドが、カレントデータベースで使用されていない場合、データベース設計者はそのフィールドに“非表示”属性を設 定し、隠してしまうことができます。クエリエディター上には、非表示ではないテーブルとフィールドだけが現れます。テーブルやフィールドを非表示にする方 法についての詳細は、テーブルプロパティおよび フィールドプロパティの属性項目を参照してください。

### 検索方法

4Dでは、強力なデータベース検索方法がいくつか提供されています。任意の検索方法を使用して、クエリを作成することができます。検索条件を満たすレコードが新しいカレントセレクションになります。

入力フォームまたは出力フォームを使用している場合、あらゆる検索方法を選択することができます。

デザインモードでは、**レコード>クエリ**メニューまたは4Dツールバー**のクエリ**ボタンメニューにて、検索に関連して4つのメニューコマンドが用意されています。各メニューコマンドは、異なるダイアログボックスやウインドウを表示し、各種オプションを提供します。これらは、実行するクエリのタイプやカレントセレクションの表示方法が異なります。

アプリケーションモードでは、クエリテーマのランゲージコマンドを使用することでこれらダイアログボックスのウインドウを表示することができます。

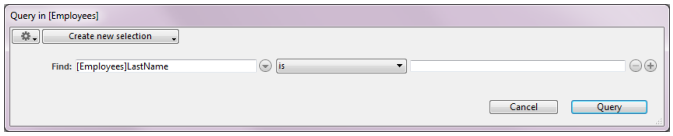
このメニューコマンドは、異なる3つの検索方法を示します:

- **検索...** (QUERY および QUERY SELECTION ランゲージコマンド): このメニューコマンドは、クエリエディターを表示します。これは汎用的な検索ダイアログボックスで、単一検索や複合検索を実行することができます。また、検索条件をディスクに保存したり、クエリをカレントセレクションに限定することもできます。
  このエディターにはフォーミュラに基づいてクエリを作成することができると上級者向けのモードも備え付けられています。(QUERY BY FORMULA と QUERY SELECTION BY FORMULA ランゲージコマンド)。例えば、フォーミュラによる検索を使い、6桁の部品番号の後ろ3桁を検索することができます。有効なフォーミュラは、ブール式 (TRUE または FALSE) を返します。
- **フォームによるクエリ...** (QUERY BY EXAMPLE ランゲージコマンド): このメニューコマンドは、カレント入力フォームを表示して検索ウインドウとして使用します。検索されるフィールドに相当するエリアに検索したい値を入力し て、クエリを指定します。複数のエリアに値を入力すると、複合検索条件を指定することができます。検索結果はカレント出力フォームに表示されます。
- **クエリ&修正...**: このメニューコマンドは、フォームによるクエリメニューコマンドと似ていますが、検索結果であるセレクションの最初のレコードがロードされ、修正可能となる点が異なります。ここで修正を行い、検索結果のレコードを閲覧しながらレコードを1件ずつ修正することができます。

# クエリエディター

クエリエディターは、単一検索や複合検索の作成に使用する多目的なエディターです。"かつ"、"または"、"以外"の論理演算子を用いて結合された複合検索を作成することができます。例えば、クエリエディターを使用して、"年齢60才以上または年収4,500,000円超"のすべての従業員を検索することができます。

検索対象として、カレントセレクションまたはテーブルの全レコードのいずれかを選択することができます。他の検索方法は、常にテーブル全体を検索します。検索条件をディスクに保存し、クエリを再度使用したい時にその条件を開くことができます。クエリエディターは最後に使用した検索条件を呼び出します。そのクエリの編集して使用することも、消去して新しいクエリを作成することもできます。カレントテーブルのフィールド、ならびにリレートテーブルのフィールドを検索することができます。また、フォーミュラを使用して上級者向けクエリを実行することもできます(フォーミュラによるクエリを参照のこと)。



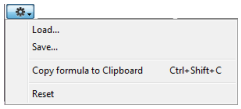

## エディターのメニュー

エディターウィンドウには編集メニューにくわえ、セレクションアクションメニューと最近の検索メニューがあります。

**注: 最近の検索**メニューはセッション中に少なくとも一つのクエリが既に実行されている場合に表示されます。

### 編集メニュー

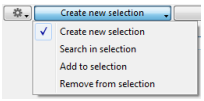
編集メニューにはクエリコードを管理するためのコマンドが含まれています。




- **読み込み...** と **保存...**: これらのコマンドは、クエリファイルのディスクから読み込み、ディスクへの保存を管理します。同じクエリを何度も実行するのであれば、ディスクにそのクエリを保存しておきたいと思う事でしょう。クエリをディスクへ保存すれば、作成するのは一度だけで済みます。その後クエリエディターを使用する際には、使用したいクエリをディスクからロードして**クエリ**をクリックして実行するだけです。
  クエリをディスクに保存するには、条件を指定した後、クエリエディター内の**保存...** をクリックします。4Dは標準のファイルを保存のダイアログボックスを開くので、そこにファイル名を入力し保存場所を指定することが出来ます。4Dクエリのファイル拡張子は".4df"です。全ての値(クエリ行、クエリアクション、フォーミュラによるクエリ)は保存されています。
  保存したクエリを読み込むためには、クエリエディターの**読み込み...** をクリックし、クエリファイル(拡張子".4df")を選択します。4Dはそのクエリをクエリエディターに読み込みます。ファイルをロードすると、それまでにクエリエディターに表示されていたクエリと置き換わります。
- **検索条件をコピー**: 編集エリア内のフォーミュラのコードをクリップボードへとコピーします。
- **リセット**: エディター内で定義された全てのクエリの行を消去します。警告:行の削除は取り消しができません。

### セレクションのアクションメニュー

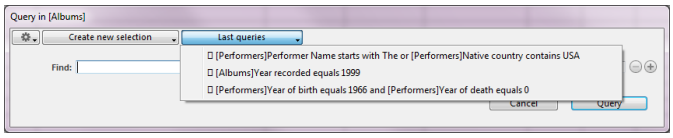
このメニューは既存のレコードのカレントセレクションに対して実行するクエリアクションを定義します。




- **セレクションの作成**(デフォルトのアクション): 4D はテーブル内の全てのレコードを検索し、元あったカレントセレクション見つかったレコードで上書きして表示します。
- **セレクションの絞り込み**: 4D は元あったカレントセレクションのレコード内のみを検索し、それを見つかったレコードで上書きして表示します。
- **セレクションに追加**: 4D はテーブル内の全てのレコードを検索し、見つかったレコードを元あったカレントセレクションに追加します。見つかったレコードのうち既にカレントセレクションの一部であったレコードに関しては表示されますが、二重に表示されることはありません。
- **セレクションから除外**: 4D はテーブル内の全てのレコードを検索し、元あったカレントセレクションから見つかったレコードを全て除外します。

### 最近の検索メニュー

このメニューは、少なくとも一つのクエリが既に実行された後にのみ表示されます。ここにはセッション中に作成された最近のクエリが表示されるので、ユーザーはいつも使っているクエリを簡単に繰り返すことが出来ます。クエリは最大10個まで保存されます。



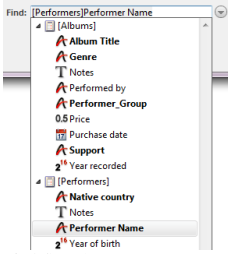

クエリに関連したセレクションのアクション(**セレクションに追加**、**セレクションの絞り込み** 等)は保存されないことに注意して下さい。

このメニューからクエリが選択されると、その詳細がビルドエリアに表示されます。その後それを直接実行したり、必要に応じて修正することができます。

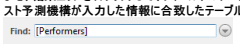
## クエリの作成

標準のクエリをエディターで作成するには、"フィールド 演算子 値"のフォームにて行を設定します:

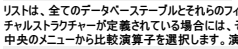
1. フィールドを指定するには編集エリアの右側にある階層リストを使用します:




また、編集エリアを右クリックして、フィールドのテーブルの最初の文字を入力することで指定することもできます("[" は入力しないでください)。テキスト予測機構が入力した情報に合致したテーブルを提示します:

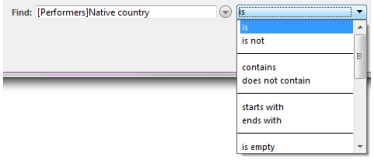



テーブルを選択した後、右矢印キー → を押してテーブルを確定させ、フィールドの一覧へ移行します。フィールドの最初の文字を入力するか、上下矢印キー ↑↓ を使用してテーブルのフィールドの一覧をスクロールして下さい:




リストは、全てのデータベーステーブルとそれらのフィールドを表示します。SET TABLE TITLES と [#cmd id="602"/] コマンドを使用してヴァーチャルストラクチャーが定義されている場合には、それも表示対象となります。

2. 中央のメニューから比較演算子を選択します。演算子の一覧は定義されたフィールドの型に応じて更新されます:




標準の比較演算子に加え、クエリエディターでは拡張された演算子と以前使用した値の型を表示するので、もっともよく使うクエリを簡単に実行

することが出来ます([#title id="1105" anchor="1127747"/]を参照して下さい)。
3. 検索したい値を入力して下さい。
テキストまたは文字フィールドではワイルドキャラクター(@)を値の最後につけることによって"前方一致"検索を行うことが出来ます。
選択したフィールドが選択リストと関連付けられている場合、4Dはリストを表示し、そこから値を選択することが出来ます。選択したフィールドがブール型フィールドであった場合、4Dは一対のラジオボタンを表示します。
4. 複数のクエリを指定したい場合、⊕ボタンをクリックして行を追加して下さい。4Dはボタンに応じて行の中身を複製します。フォーミュラを使用したクエリ行を追加したい場合には、**Alt** (Windows) または **Option** (OS X) を押しながらボタンをクリックして下さい。この点については フォーミュラによるクエリ セクションに詳細な説明があります。
5. 指定したい接続詞(**And**, **Or**, **除外**)を選択して下さい。
デフォルトでは4Dは追加された行の隣にAnd接続詞を設定します。
6. 1から3までの手順を繰り返して新しい条件を指定します。
複合クエリを作成した場合、4Dはクエリエディター内に表示されている順(上から下)に各行のクエリを実行していきます。接続詞には優先順位はありません。つまり、AndがOrより優先される、といったようなことはないという事です。そのため2行以上にわたるクエリを使用した場合は、条件の入力する順番によってクエリの検索結果が変化する可能性があります。
複合クエリを作成している途中、行やフィールドや演算子をクリックしたり、新しい値を入力したりすることにより既存のクエリ部分を修正することもできます。
行のとなりにある ⊖ ボタンをクリックすることにより、クエリの行を削除することもできます。行の削除は取り消しができないので十分注意して下さい。
7. セレクションアクションメニューからクエリの結果の使用方法を以下から選択して下さい。**セレクションの作成**、**セレクションの絞り込み**、**セレクションに追加**、**セレクションから除外** (前のセクションを参照して下さい)。
8. クエリをディスクに保存するためには、**編集**メニューから**保存...**を選択して下さい。
9. クエリをクリックして検索を実行して下さい。

**比較演算子**

クエリを作成する際、指定した値とデータベースの内容とを比較する方法を4D に対して指示しなければなりません。例えば名字が"佐藤"と等しいというクエリの場合は、等しいという比較演算子を使用します。これは4Dに対して、名字 フィールドの値と文字列である"佐藤"とを比較するように指示します。

英数字タイプの値に関する比較では、大文字小文字が区別されません。名字が"Smith"である人を検索すると、"smith"や"SMITH"、"sMith"などを含むレコードが検出されます。

**含む**および**含まない**比較演算子を用いたクエリは、常にシーケンシャル検索になります。**キーワード** 演算子は文字型とテキスト型のフィールドに対してのみ使用可能です。この型のクエリについての詳細は4D *Language Reference* マニュアルの比較演算子を参照して下さい。

それぞれの型において使用できる演算子は以下の通りです:

| 演算子 | 文字/テキスト | 日付 | 時間 | ブール | 数値 | ピクチャ | 詳細 |
|---|---|---|---|---|---|---|---|
| 空の文字列/=空 | x | | | | | x | フィールドにはデータがありません。 |
| ≠空の文字列/≠空 | x | | | | | x | フィールドにデータはあります。 |
| と等しい | | | | | x | | 数値用の標準の演算子です。 |
| >= | | | | | x | | |
| > | | | | | x | | |
| <= | | | | | x | | |
| < | | | | | x | | |
| ≠ | | | | | x | | |
| 偽(false)である | | | | x | | | ブール用の標準の演算子です。 |
| 真(true)である | | | | x | | | |
| = | x | | x | x | | | フィールドは入力された値と完全に一致します。 |
| ≠ | x | | x | x | | | フィールドは入力された値と異なります。 |
| >= | x | | x | x | | | フィールドの値は、入力された値と一致するかまたは大きいです(*) |
| > | x | | x | x | | | フィールドの値は入力された値を超えます(*) |
| <= | x | | x | x | | | フィールドの値は、入力された値と一致するかまたは小さいです(*) |
| < | x | | x | x | | | フィールドの値は入力された値未満です(*) |
| 範囲指定 (以上以下) | | | x | | | | 最初の日付は二番目以前の日付である必要があります。入力された日付も含め、間にある日付のフィールドを検索します |
| 範囲指定 | | | x | | | | 最初の日付は二番目以前の日付である必要があります。入力された日付は含めずに検索します |
| = 今日 | | | x | | | | カレントの日付が表示されている |
| = 昨日 | | | x | | | | 一日前の日付が表示されている |
| = 今週(月・四半期・年)中 | | | x | | | | とり得る値 :- 週(日-土) -週(月-日) -週(月-金) -月 -四半期 -年 これらの値は、カレントの日付を基準として算出されます。 |
| = 昨週(月・四半期・年)中 | | | x | | | | |
| = 翌週(月・四半期・年)中 | | | x | | | | |
| = 時間(分・秒)以内の過去 | | | | x | | | とり得る値 : -時間 -分 -秒 これらの値は、カレントの時刻を基準に算出されます。 |
| = 時間(分・秒)以内の未来 | | | | x | | | |
| 範囲指定(以上以下) | x | | | x | x | | フィールドの値は入力された値の間です(入力された値も含みます)。(*) |
| 範囲指定 | x | | | x | x | | フィールドの値は入力された値の間です(入力された値は含まれません)。(*) |
| = | | | | x | | | とり得る値 : -時間 -分 -秒 |
| ≠ | | | | x | | | |
| >= | | | | x | | | |
| > | | | | x | | | |
| <= | | | | x | | | |
| < | | | | x | | | |
| 前方一致 | x | | | | | | テキスト用の標準の演算子です。 |
| 後方一致 | x | | | | | | |
| 含む | x | | | | | | |
| 含まない | x | | | | | | |
| キーワード | x | | | | | x | 入力されたキーワードを検索します。「すべて」(を含むフィールドを検索する)と「どれか」(を含むフィールドを検索する)のどちらかを指定することが出来ます。 |
| キーワード(含まない) | x | | | | | x | |
| <= | | | | | | x | ピクチャのサイズに基づいて検索します。単位はバイト、KB、MB、GBから選択することが出来ます。 |
| >= | | | | | | x | |

(*) 文字列の場合、クエリはアルファベットに基づいています(a<bとみなされます)。例えば、「名前が "don" よりも後である」というタイプのクエリを実行した場合、検索結果には Donna、Don、Juan、Smith、、、等が含まれますが、AlvesやDominickはヒットしません。

**ワイルドカード記号 (@)**

4Dでは、クエリを指定しやすくするためにワイルドカード記号 (@) が用意され、文字またはテキストタイプのフィールドに関する検索において、1桁以上の文字を置き換えることができます。例えば、あるフィールドで"佐藤四太郎"という名前をすべて探し出す場合、いくつかの方法を用いて検索を指定することができます:

| 検索条件: | 検出される値 |
|---|---|
| 佐藤@ | "佐藤"で始まるすべての値 |
| @太郎 | "太郎"で終わるすべての値 |
| 佐@太郎 | "佐"で始まり、"太郎"で終わるすべての値 |
| @太@ | "太"を含む全ての値 |

**注:** ワイルドカード記号は検索値の最後に配置されている場合にかぎり、"キーワードを含む"と組み合わせて使用することも可能です。詳細については4D *ランゲージリファレンス*マニュアルのOpen external windowを参照してください。

**単一検索と複合検索**

1つまたは複数のフィールドを検索することができます。1つのフィールドに基づくクエリは、"単一検索"と呼ばれます。例えば、名字が"佐藤"に等 しいという検索は単一検索です。単一検索を実行すると、4Dはデータベースを検索する際に、1つのフィールドの内容を調べます。

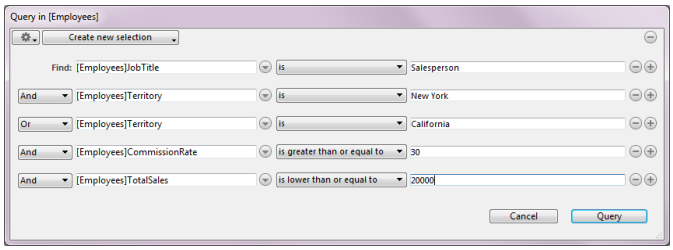
複数のフィールドに基づく検索は、"複合検索"と呼ばれます。複合検索を行う場合は、論理演算子を使用して個々のクエリを組み合わせます。論理演算子により、それぞれのクエリの結果を組み合わせる方法を4Dに指示します。3種類の論理演算子があります:

- **かつ**: この演算子は、2つの条件を同時に満たすレコードをすべて検索します。例えば、"技術部門で働き、かつ給与が5,000,000円以上の従業員をすべて検索する"というクエリは、給与が5,000,000円以上である技術部門の従業員のレコードだけを探し出します。
- **または**: この演算子は、2つの単一検索のいずれかの条件を満たすレコードを検索します。例えば、"技術部門で働く従業員、または給与が5,000,000円以上で ある従業員をすべて検索する"というクエリは、技術部門で働く従業員、および働く部門とは関係なく給与が5,000,000円である従業員をすべて探し出 します。
- **以外**: この演算子は、"~ではない"と同じです。例えば、"給与が5,000,000円以上である従業員以外の技術者をすべて検索する"というクエリは、給与が5,000,000円以上である技術者を除外します。

複数の条件を組み合わせたクエリを実行するためには、必要な条件の数の分だけ行を追加ボタン⊕をクリックして下さい。

論理演算子を使用すると、"東京または大阪に所属し、かつ販売高が2,000,000円より少なく、かつ歩合が30パーセントを越える販売員を検索す

る"と いうような複合検索を作成することができます。次の図は、このクエリがクエリエディター上で指定されている様子を示しています:




このクエリが実行されると、4Dは低い販売高売上に対して高い歩合を得ている、東京または大阪の販売員をすべて検索します。比較演算子と論理演算子を使用するこの他の例題は、各検索メソッドにおいて紹介されています。

## フォーミュラによるクエリ

フォーミュラによるクエリを使用し、計算結果やデータ操作に基づいたレコード検索を行うことができます。作成する式は、各レコードに関してその結果が"True (真) "または"False (偽) "にならなければなりません。フォーミュラには、4Dのコマンドやランゲージのすべての関数、ならびに開発者が特に指定した任意のプロジェクトメソッドを使用することができます。

フォーミュラによるクエリコマンドは、次のような操作に関係するクエリを作成する際に役立ちます:

- 文字列に関する操作や評価の実行
- 日付演算の結果による検索
- 算術計算による検索

フォーミュラによるクエリエディターの使用例をいくつか紹介します:

- 次のフォーミュラは、電話番号フィールドの最後の7桁が"2524444"と等しいレコードを検索します:
  `Substring([Emp]Phone number;4;7)="2524444"`
- 次のフォーミュラは、年度とは関係なく、本日生まれの人を検索します:
  `(Day of(Current date)=Day of([Emp]Birthdate)) & (Month of(Current date)=Month of([Emp]Birthdate))`
- 次のフォーミュラは、年間売上高を生計費で割り、その計算結果が1,000を超えるレコードを検索します:
  `([Stats]Annual Sales/[Stats]CostOfLiving)>1000`

しかし、次のフォーミュラ:

```
[Stats]Annual Sales/[Stats]CostOfLiving
```

の場合は、"True (真) "または"False (偽) "ではなく、数値を返すため正しくありません。

フォー ミュラは1行の論理式でのみ作成することができます。つまり、キーボード上の Enter キー (Macintosh では Return キー) を押して、2行目を記述することはできません。しかし、命令文が長すぎる場合には、編集エリア内で行を折り返して表示します。1行を超えるフォーミュラを 使用する必要がある場合には、それをプロジェクトメソッドとして作成し、フォーミュラの中でこのメソッドを使用してください。

### フォーミュラによるクエリの実行

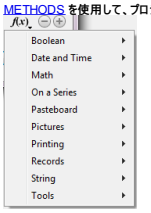
フォームによるクエリは標準のクエリエディターの中に統合されました。フォームによるクエリとは高度なクエリモードです。

フォームによるクエリを定義するためには、まず クエリエディター を表示させ、まずラインを追加ボタン を **Alt+ クリック**(Windows) または**Option+ クリック**(Mac OS) します。

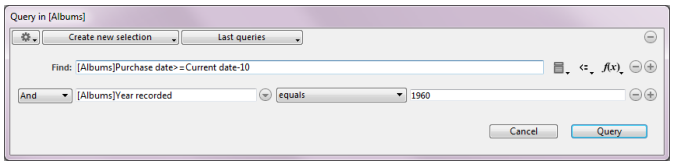
**注:** 単一行のフォーミュラによるクエリを実行したい場合、デフォルトで追加されている最初の行を削除して下さい。

追加された行にはメニューが増えています:

- : をクリックすると、フォームによるクエリで使用できる全てのデータベースのテーブルとフィールドの一覧の階層リストが表示されます。
- : をクリックすると、フォームによるクエリで使用できる全ての演算子の一覧の階層リストが表示されます。

  
- : をクリックすると、フォームによるクエリで使用できる全ての 4D ファンクションの一覧の階層リストが表示されます。SET ALLOWED METHODS を使用して、プロジェクトメソッドをこのリストに追加することもできます。
  
  

この新機能をを通じて、フォームによるクエリの条件を標準のクエリの条件と組み合わせて使用することが出来るようになりました:




また、こうしてこのダイアログボックスへと統合されたことにより、フォームによるクエリは以下の点において標準のクエリと同じ機能を持つようになりました:

- セレクションのアクションのメニュー(セレクションのアクションメニュー)を使用してクエリアクションを選択できるようになりました。
- 直前の検索(最近の検索メニュー)の一覧の中から実行済みのクエリを再度選択できるようになりました。
- ファイルメニューを通して保存や読み込みができるようになりました。(編集メニュー を参照して下さい)

## フォームによるクエリ

フォームによるクエリオプションは、数多くのクエリを実行するための便利な方法です。このタイプのクエリでは、カレント入力フォームを用いて、検索する値を入力します。カレントテーブルのフィールドだけが検索対象となります。フォームによるクエリは、インデックスによるクエリとシーケンシャルクエリの両方を実行します。

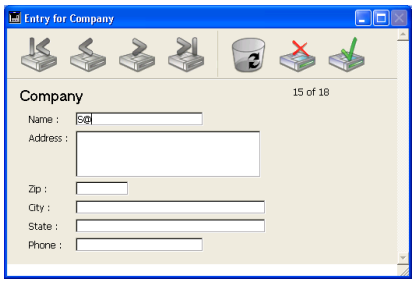
下図のフォームによるクエリウインドウは、ひとつの例です。カレント入力フォームが常にフォームによるクエリウインドウとして用いられます。フォームによるクエリウインドウの外観を制御するには、カレント入力フォームを変更します。

フォームによるクエリでは比較演算子を使用することができます。例えば、比較演算子"="を使用するには、検索する値を対応するフィールドに入力します。その他の比較演算子が必要な場合は、次の記号のいずれかを値の前に付加します:

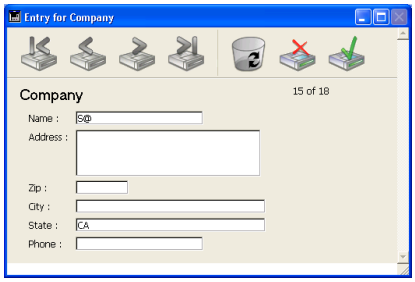
| 比較 | 演算記号 | 例 |
|---|---|---|
| 等しくない | # | #Marketing |
| より大きい | > | >30000 |
| 以上 | >= | >=30000 |
| より小さい | < | <30000 |
| 以下 | <= | <=30000 |

検索する値の後にワイルドカード記号 (@) を指定すると、"先頭一致"クエリを要求することができます。

下図の例では、"S"で始まる名称の会社を検索するために、フォームによるクエリを使用しているところを示しています。




また、複数のフィールドに値を入力して、複合検索を作成することも可能です。複合検索を作成する場合、論理演算子"かつ"を使用するものとみなされます (例: "名称がSで始まり、かつCA (カリフォルニア州) にある") 。次のフォームによるクエリウインドウは、複合検索の例を示します。




フォームによるクエリを使用するには、次の手順に従ってください:

1. デザインモードで**レコード**メニューから**クエリ>フォームによるクエリ...**を選択します。
   または
   ツールバーにある**クエリ**ボタンのサブメニューから**フォームによるクエリ...**を選択します。
   4Dはフォームによるクエリウインドウとしてカレントテーブルの入力フォームを表示します。**登録**ボタンと**キャンセル**ボタンだけが使用可能です。
2. 検索する値をフィールドに入力します。
   例えば、名字が"佐藤"である人のレコードをすべて検索するには、名字フィールドに"佐藤"と入力します。
   比較演算子を使用する場合は、値の前に比較演算子を付加します。"先頭一致"検索を実行するには、値の後に"@"記号を付加します。
   **注:** フォームによるクエリでは、"キーワードを含む"検索は使用できません。Enter a value to search for in a field.
3. 複合検索を実行するには、他のフィールドにも値を入力します。
   複数のフィールドに値が入力されると、4Dは論理演算子として"かつ"を使用します。
4. **登録**ボタンをクリックするか、テンキー上の **Enter** キーを押して、クエリを実行します。
   クエリをキャンセルするには、**キャンセル**ボタンをクリックするか、または **Escape** キー (Mac OS では **Esc** キー) を押します。
   検索結果はカレント出力フォームに表示されます。

## クエリ&修正

クエリ&修正コマンドは、レコードを素早く探し出して修正することを目的としています。フォームによるクエリと同様に、クエリウインドウとしてカレント入力フォームを使用します。また、クエリの指定方法も、フォームによるクエリとまったく同じです。

1件のレコード、またはレコードセレクションが見つかると、4Dは修正のために一番目のレコードを入力フォームに表示します。レコードを登録すると、出力フォームに戻ります。

- クエリ&修正を使用すると、見つかったレコードはカレント出力フォーム上に表示されます。新しいセレクション内の全レコードが同時に表示されます。
- クエリ&修正を使用すると、新しいセレクションのカレントレコードは、カレント入力フォームにも表示されます。したがって、値を希望通りに直接書き換えることができます。新しいセレクションに複数のレコードがある場合は、入力フォームのナビゲーションボタンをクリックし、レコード間を移動して1つずつ値を修正することができます。

クエリ&修正を使用するには、次の手順に従ってください:

1. デザインモードで**レコード**メニューから**クエリ>クエリ&修正...**を選択します。
   または
   ツールバーにある**クエリ**ボタンのサブメニューから**クエリ&修正...**を選択します。
   4Dはクエリウインドウとして、カレント入力フォームを表示します。**登録**ボタンと**キャンセル**ボタンだけが使用可能です。
2. フォームによるクエリと同様に検索条件を設定します。
   詳細についてはフォームによるクエリを参照してください。
3. **登録**ボタンをクリックするか、テンキー上の **Enter** キーを押して、クエリを開始します。
   4Dはクエリを実行してカレントセレクションを設定し、検索条件を満たす最初のレコードを入力フォーム上に表示します。
4. カレントセレクション内の別のレコードに移動するには、ナビゲーションアイコンをクリックします。

## レコードの並び替え


レコードの並び替え

並び替えエディター

## レコードの並び替え

並び替え処理は、テーブルの値に基づいてレコードを並べ替えます。一般的には、次のような場合にレコードを並べ替えます:

- 特定の順序で画面上にレコードを表示するため。
- レポートやラベルを印刷する前。
- データをグラフ化する前。

新規データベースにデータを入力する際、4Dはデータの入力順または読み込み順にレコードを保存します。出力フォーム上にレコードを一覧表示したり、レコードを印刷する時も、この順番で表示されます。しかし、レコードを別の順番で表示したい場合が少なくありません。例えば、レポート上で名簿リストを50音順に並べ替えたい場合もあるでしょう。この場合は、名字フィールドで並べ替えると、名字が50音順 (正確には、シフト JIS コード順) に並びます。

入力フォームまたは出力フォームを使用して、並び替えを実行することができます。入力フォームから並び替えを行うと、新しい並び順における先頭レコードが入力フォームに表示されます。出力フォームで並べ替えを行うと、並べ替えられたレコードは出力フォーム上に表示されます。

4Dはインデックスによる並び替えを非常に高速に実行します。並べ替えるフィールドが1つだけであり、かつそのフィールドがインデックス定義されている場合、4Dはそのインデックスを利用します。

並び替えエディターを使用すると、並び替え条件を手動で指定して実行することができます。

アプリケーションモードでは、ORDER BY やORDER BY FORMULA コマンドを使用して並べ替えを直接実行することができます。また、同コマンドによって並び替えエディターを表示させることも可能です。

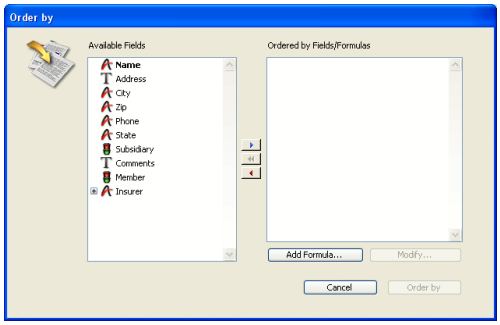
デザインモードにおいては、**レコード**メニューに**並び替え...**コマンドが提供されています。

並べ替えはカレントセレクションのレコードが表示、または印刷される順序を変更します。この並び替えは一時的なものであり、カレントセレクションに対してのみ適用されます。したがって、データベースに保存されているレコードの順序には影響を与えません。通常、並び替えは検索の直後、またはレポートやラベルの印刷直前に使用されます。

## 並び替えエディター

### 説明

デザインモードにおいて、**レコード**メニューまたはツールバーにある**クエリ**ボタンのサブメニューより**並び替え...**を選択します。




並び替えエディターには、次のエリアおよびボタンがあります:

- **フィールドリスト**: このエリアには、カレントテーブルのフィールドが階層リストで表示されます。インデックスフィールドは太字で示されます。サブテーブルやリレートテーブルのフィールドに基づく並び替えも行えます (リレートが自動である場合) 。サブテーブルやリレートテーブルのフィールドを使用するには、プラス (+) 記号 (Windows) 、または矢印 (Macintosh) をクリックして外部キーフィールドを展開し、リレートしているフィールドを表示します。
  **注:** 非表示設定されたテーブルやフィールドは並び替えエディターに表示されません。テーブルやフィールドの非表示設定に関する説明は、テーブルプロパティおよびフィールドプロパティの"属性"項目を参照してください。
- **並び替え条件エリア**: このエリアには、並び替えフィールド、並び替えフォーミュラ、および各並び替えの方向が表示されます。このエリアの右側にある矢印を使用して、昇順並び替えまたは降順並び替えを指定します。
- **フォーミュラ追加...**ボタン: このボタンを使用すると、並び替え条件の1つとしてフォーミュラを作成することができます。計算結果やフィールドの一部など、フィールド以外のものを並び替えたい場合に、フォーミュラを使用します。例えば、部品の不良品率や部品番号フィールドの下4桁を並べ替えることができます。
- **修正...**ボタン: このボタンをクリックすると、選択した並び替え条件がフォーミュラエディターに表示されます。選択した条件がフォーミュラである場合、そのフォーミュラが編集のために表示されます。また、条件がフィールドである場合、そのフィールド名がフォーミュラエディターの編集ウインドウに表示されます。
- **ボタンパネル**: ボタンパネルには、並び替えリストからフィールドを追加したり、削除するためのボタンがあります。フィールドはドラッグ&ドロップで追加することも可能です。
- **キャンセル**ボタン: このボタンを使用し、並び替えを取り消して使用中のフォームに戻ります。
- **並び替え**ボタン: このボタンを使用して、並び替えを実行します。クリックすると4Dは並び替えを実行し、並び替えられたレコードの一覧を表示します。入力フォームから並び替えを行った場合には、新しい並び順における先頭レコードが入力フォームに表示されます。

**注:** クイックレポートエディターでは、クイックレポートに表示されるレコードを並べ替えることができます。

### 並び替えレベル

最大30種類のフィールドやフォーミュラに基づいてレコードを並べ替えることができます。並び替える各フィールドやフォーミュラは、"並び替えレベル"と呼ばれます。例えば、名字フィールドと名前フィールドの2レベルでの昇順並び替えにより、次のようなリストが出力されします:

アオキイサム
アオキタイジ
アオキヨウコ
　[...]
ワタナベタカシ
ワタベヒロシ

フィールド内容を並べ替える際、4Dは大文字と小文字を区別しません (つまり、"Aoki"="aoki")。また、発音記号による区別も行いません (つまり、"Aá"="Aa")。ただし、インターナショナル版の4Dを使用しており、お使いのオペレーティングシステムで発音記号が区別されている場合 (つまり、"Aá"≠"Aa")、並び替え操作でも発音記号が考慮されます。

### 並び替え条件エリア

並び替え条件エリアにおいて、並び替え条件の順番を変えたり、削除したりすることができます。

並び替え条件の順序を変更するには、ドラッグ&ドロップ操作を使います:

1. 並び替え条件エリアにて順序変更したいフィールドやフォーミュラをクリックし、そのままドラッグして上位または下位に移動します。
   移動させた条件はドロップ先の一つ上に挿入されます。

並び替え条件の削除:

1. 並び替え条件エリアにて削除したいフィールドまたはフォーミュラを選択して反転表示させます。
2. **Backspace** キー (Macintosh では **Delete** キー) を押します。
   または
   削除ボタンをクリックします。
   並び替え条件エリアから選択した条件が削除されます。

並び替え条件の全削除:

1. すべてを削除ボタンをクリックします。
   並び替え条件エリアからすべての条件が削除されます。

### 昇順と降順

並び替えを実行する各フィールドやフォーミュラに対し、昇順または降順のいずれかを指定することができます。"あ"から"ん"、または最小から最大へ向かう並び替えは"昇順並び替え"と呼ばれます。これとは逆方向、つまり最大から最小、最後から最初、"ん"から"あ"への並び替えは"降順並び替え"と呼ばれます。

複数レベルで並び替えを行っている場合、昇順並び替えと降順並び替えを自由に組み合わせることができます。つまり、複数レベルの並び替えには、フィールドやフォーミュラ、昇順並び替えと降順並び替えを混在させることができます。

並び替え方向の変更:

1. 並び替え条件エリアにて、条件の右に表示されている矢印をクリックします。

### フォーミュラに基づいて並べ替える

並び替えにはフィールドだけでなく、フォーミュラも使用できます。例えば、次のフォーミュラは誕生日フィールドの月で並べ替えます。

```
Month of([Children]Birth Date)
```

並び替え用のフォーミュラを作成するには、次の手順に従ってください:

1. 並び替えエディターの**フォーミュラ追加...**ボタンをクリックします。
   4Dはを表示します。
   フォーミュラエディターを使用して、並べ替える値を返すフォーミュラを作成します。あらゆるデータタイプの値を返すフォーミュラを使用することができます。フォーミュラエディターの使用方法に関する詳細については、を参照してください。
2. フォーミュラの作成が完了したら、**OK**ボタンをクリックします。
   4Dは並び替え条件エリアにフォーミュラを表示します。フォーミュラとともに、並び替え方向を示す矢印も表示されます。

フォーミュラを修正したり、並び替え条件エリアに表示されているフィールドを使ってフォーミュラを作成するには、任意のフォーミュラまたはフィールドを選択して**修正...**ボタンをクリックするか、並び替え条件エリアのフォーミュラ名やフィールド名をダブルクリックします。

## フォーミュラエディター

フォーミュラエディターの使用場面
フォーミュラエディター

## フォーミュラエディターの使用場面

フォーミュラエディターは4Dデータベースの様々な場面で使用できます:

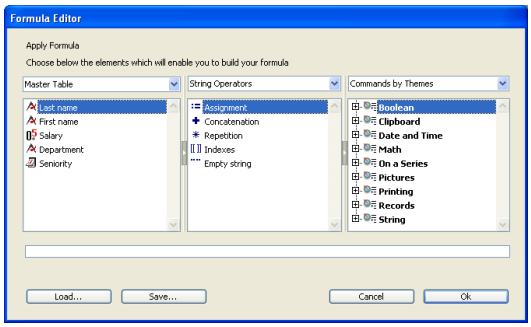
- 一連のレコードに対してグローバル更新を行う時 (レコードを編集するの"グローバル更新"項目参照)
- フォーミュラを使用して並び替えを行う時 (並び替えエディターの"フォーミュラに基づいて並び替える"項目参照)
- クイックレポートのカラムにフォーミュラを追加する時 (クイックレポートにフォーミュラを関連付ける参照)

アプリケーションモードでは、EXECUTE FORMULAコマンドを使用してフォーミュラを直接実行することができます。また、EDIT FORMULAコマンドによってフォーミュラエディターを表示させることも可能です。

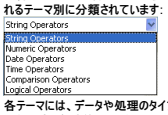
**注:** フォーミュラはデータベースのレコードに対しても使用することが出来ます。この場合、クエリエディターの高度な機能を使用し、フォーミュラエディターは使用しません(フォーミュラによるクエリを参照して下さい)。

## フォーミュラエディター

フォーミュラエディターでは、フォーミュラを作成するために数々のショートカットが提供されます。フィールド名、演算子、コマンド、および任意のプロジェクトメソッドをクリックすると、それがフォーミュラに追加されます。項目をクリックすると、自動的にその項目が編集エリアに表示され、このエリアで標準のカット、コピー、ペースト機能を使用して修正を行うことができます。また、編集エリアに項目を直接入力したり、ドラッグ&ドロップを用いて項目を追加することも可能です。




フォーミュラエディターには次のようなエリアがあります:

- **テーブルとフィールドのリスト**: このエリアには、テーブルのフィールドが表示されます。リストの上にあるメニューを使用すると、表示されるフィールドを設定することができます。カレントテーブルのフィールド、リレートテーブルのフィールド、または全テーブルのフィールドが使用可能です。
  **注:** 非表示設定されたテーブルやフィールドはリストに表示されません。テーブルやフィールドの非表示設定に関する説明は、テーブルプロパティおよびフィールドプロパティの"属性"項目を参照してください。
- **演算子リスト**: 演算子リストを使用すると、フォーミュラで使用する演算子を選択することができます。演算子は、リストの上のメニューに表示されるテーマ別に分類されています:
  
  
  各テーマには、データや処理のタイプに対応する使用可能な演算子が表示されます。例えば、代入演算子":="は、すべてのデータタイプで使用可能です。各演算子の説明は、次の項目を参照してください。
- **コマンドリスト**: コマンドリストには、フォーミュラ内で使用できる4D関数や、開発者が認可したプロジェクトメソッドが納められています。リスト上部にあるメニューを使用すると、テーマ別またはアルファベット順にコマンドを表示できます。このメニューに表示される4Dコマンドの説明は、4D ランゲージリファレンスマニュアルを参照してください。
  フォーミュラで使用されるプロジェクトメソッドは、4Dコマンドである SET ALLOWED METHODS を用いて宣言しておかなければなりません。ただし、データベースのデザイナーと管理者にはあらかじめ、フォーミュラエディターにおいてすべてのコマンドとユーザーメソッドへのアクセス権が与えられています。このアクセス権はすべてのユーザーに対して無効に設定することもできます。これらのオプションはデータベース環境設定のセキュリティページにて設定できます。

### フォーミュラの演算子

次に、フォーミュラエディターで利用できる各種演算子について簡単に説明します。これらの演算子で実行できる機能に関する詳しい説明は、4D ランゲージリファレンスマニュアルの演算子の章を参照してください。

- **文字列演算子**
  A と B は文字列、N は数値です。

| 演算子 | 使用方法 | 説明 |
|---|---|---|
| := 代入 | A:=B | 値BをAに代入する |
| + 連結 | A+B | ABを返す |
| * 反復 | A*N | 値AをN回繰り返す |
| [[ ]] インデックス | [[A]]N | 値AのN桁目を返す |
| "" 空の文字列 | "" | 一組のダブルクオートを挿入する |

- **数値演算子**
  XとYは数値です。

| 演算子 | 使用方法 | 説明 |
|---|---|---|
| := 代入 | A:=B | 値BをAに代入する |
| + 加算 | X+Y | X+Yの結果を返す |
| - 減算 | X-Y | X-Yの結果を返す |
| * 乗算 | X*Y | X*Yの結果を返す |
| / 除算 | X/Y | X/Yの結果を返す |
| // 整数値を返す除算 | X//Y | X/Yの結果の整数部分を返す (XとYは整数でなければならない) |
| % モジュロ | X%Y | X/Yの結果の余りを返す |
| ^ 指数 | X^Y | XのY乗を返す |

**注:** モジュロ演算子 (%) は倍長整数の範囲内 (-2^31から +2^31-1まで) の数値に関して有効数字を返します。この範囲外の数値のモジュロ演算を実行するには、Modコマンドを使用してください。

- **日付演算子**
  D1と D2は日付、N は数値です。

| 演算子 | 使用方法 | 説明 |
|---|---|---|
| := 代入 | D1:=D2 | 値D2をD1に代入する |
| + 日付の加算 | D1+N | D1+N日の結果を返す |
| - -日付の減算 | D1-D2 | D1からD2までの日数を返す |
| | または D1-N | D1-N日の結果を返す |
| !//! 空の日付 | !00/00/00! | 空の日付を挿入する |

- **時間演算子**
  H1と H2は時間、N は数値です。

| 演算子 | 使用方法 | 説明 |
|---|---|---|
| := 代入 | H1:=H2 | 値H2をH1に代入する |
| + 加算 | H1+H2 | H1+H2時間の結果を返す |
| | または H1+N | H1+N秒の結果を返す (夜の12時からの経過時間を秒数で表わす) |
| - 減算 | H1-H2 | H1-H2の結果を返す |
| | または H1-N | H1-N秒の結果を返す (夜の12時からの経過時間を秒数で表わす) |
| * 乗算 | H1*N | H1*Nの結果を返す (夜の12時からの経過時間を秒数で表わす) |
| / 除算 | H1/N | H1/Nの結果を返す (夜の12時からの経過時間を秒数で表わす) |
| // 整数値を返す除算 | H1//N | H1/Nの結果の整数部分を返す (夜の12時からの経過時間を秒数で表わす) |
| % モジュロ | H1%N | H1/Nの結果の余りを返す |
| ?::? 空の時間 | ?00:00:00? | 空の時間を挿入する |

- **比較演算子**
  Z1と Z2は、文字列、数値、日付、または時間タイプです。

| 演算子 | 使用方法 | 説明 |
|---|---|---|
| = 等号 | Z1=Z2 | Z1=Z2であればTrueを返す |
| # 不等号 | Z1#Z2 | Z1#Z2であればTrueを返す |
| > より大きい | Z1>D | Z1>Z2であればTrueを返す |
| >= より大きいまたは等しい | Z1>=Z2 | Z1>=Z2であればTrueを返す |
| < より小さい | Z1<Z2 | Z1<Z2であればTrueを返す |
| <= より小さいまたは等しい | Z1<=Z2 | Z1<=Z2であればTrueを返す |

- **論理演算子**
  B1と B2はブールタイプでなければなりません (TRUE 式または FALSE 式)。

| 演算子 | 使用方法 | 説明 |
|---|---|---|
| & かつ | B1 & B2 | B1がTrueでB2がTrueであれば、Trueを返す |
| ¦または | B1 ¦B2 | B1がTrueかB2がTrueであれば、Trueを返す |

# クイックレポート


はじめに

クイックレポートの基礎

クイックレポートエディター

レポートのグラフィック属性を変更する

レコードの並び替え

集計計算の追加

表示フォーマットの設定

行や列の表示と非表示

ページヘッダーとページフッターを追加する

クイックレポートの生成

4D コードを生成する

## はじめに

データ管理における最も重要な処理の1つにレポートの生成があります。クイックレポートエディターは、4Dでレポートを設計するために利用できる2 つのツールのうちの1つです。デザインモードおよびアプリケーションモードにてクイックレポートエディターを使用し、非定型のレポートを作成することがで きます。

もうひとつのツールはフォームエディターですが、これはデザインモードでレポートを設計するために使用します (後からアプリケーションモードで修正可能)。複雑なデザインやプログラムによる処理を必要とするレポートの設計には、出力フォームを使用されることをお 薦めします。詳しくは一覧フォームとレポートを参照してください。

クイックレポートエディターを使用すると、次のような操作を実行できます:

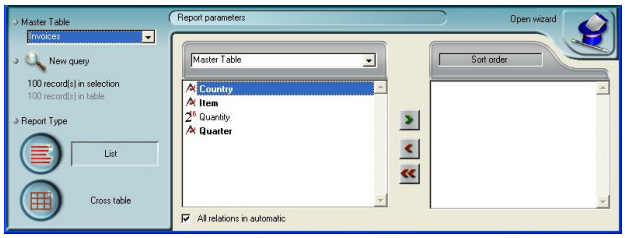
- レコードリストを作成します。
- ブレークエリアを作成します。
- クロスタブ (クロス集計表) レポートを作成します。
- 合計計算を実行します。
- レポートにおけるフォントやスタイルを修正します。
- セル単位で罫線や背景色を指定します。
- クイックレポートデザインをディスクへ保存したり、ディスクから開いたりします。
- HTMLまたはテキストファイル、4DViewエリア、印刷またはディスクへ保存など、さまざまな出力タイプを選択します。

クイックレポートエディターはクイックレポートテーマのコマンドによって制御できます。

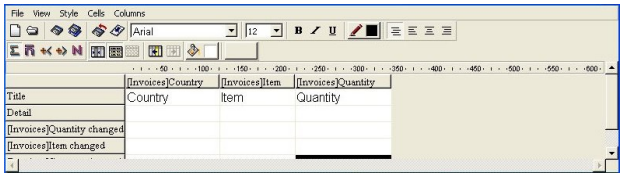
### クイックレポートエディター

エディターは、2つの主要エリアから構成されています:

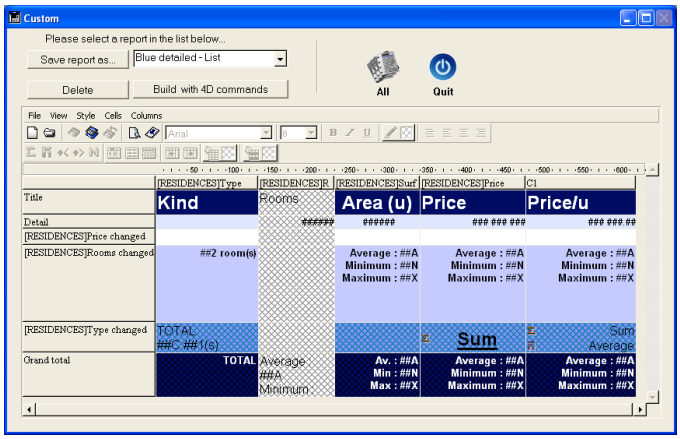
クイックレポートの**データエリア**では、作成するレポートのタイプや統合するデータの種類を指定します。また、クイックレポートウィザードへアクセスするためにも使用できます:




もう一つは、実際の**クイックレポートエリア**です:




上の図は、フォームに挿入された状態のクイックレポートエリアを表示しています。このエリアにおいて実際のカラムや行の内容が作成され、並び替え順などの定義が行われます。また、このエリアには、各種ツールバーや独自のメニューバーとコンテキストメニューが含まれます。
例として、このエリアは下図のようなフォーム上に配置することができます:




### リストモードとクロステーブルモード

クイックレポートエディターでは、2種類の処理モードが使用可能であり、リストとクロステーブルという2種類のレポートを生成することができます。

- **リストモード**
  このモードはデフォルトのモードです。通常レポートはこのモードにおいて、累計が行われるブレークレベルを伴うリスト形式でレコードを表示します。一般的なリストタイプのクイックレポートを次に示します:

| 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 |

- **クロステーブルモード**
  このモードを使用すると、二次元の表形式でレポートを表示することができます。このモードは実際に2つのデータソースに由来する、カテゴリに分割可能なデータをレポート化する場合に便利です。
  例えば、クロステーブルモード形式を使用すると、四半期ごとの製品タイプ別販売金額を表にして表わすことができます。典型的なクロステーブルタイプのクイックレポートを次に示します:

| | AV Preamplifier | Power module | Remote control | Line Total |
|---|---|---|---|---|
| Q1 | 34 | 29 | 39 | 102<br>10526316 |
| Q4 | 48 | 64 | 21 | 133<br>66666667 |
| Q3 | 49 | 68 | 40 | 157<br>48148148 |
| Q2 | 64 | 74 | 47 | 185<br>66666667 |
| Grand total | 195<br>5.90909090909<br>1 | 235<br>5.875<br>1 | 147<br>5.44444444444<br>1 | 577<br>5.77<br>1 |

表示しようとするデータを含むデータソースがカテゴリー別に分かれており、それらのカテゴリーが実際には各々別のデータソースに属している場合、このモードが便利です。

**クイックレポートウィザード**

手動でクイックレポートを作成する際にウィザードを利用すると、レポート作成のためのガイドを受けることができます。
ウィザードを起動するには、クイックレポートエディターの**ウィザードを開く**ボタンをクリックします:

すると、クイックレポートエディターウインドウのデータエリアがウィザードのページに置き変えられます。レポートの作成は細かいステップを経て実行され、ウインドウの右側には解説が表示されます。ウィザードを使用してクイックレポートを作成するには、画面上に表示される指示に従ってください。

レポートの作成が完了した時点で、ウィザードではレポートに対応する4Dコードを生成する機能が提供されます。詳細は、4D コードを生成するを参照してください。

ウィザードを終了するには、手動モードボタンをクリックして手動モードに戻ります:

# クイックレポートの基礎

### 新規クイックレポートを作成する

クイックレポートは、デザインモードまたはアプリケーションモードにてエディターを使用して、作成することができます。また、Reportタイプのプラグインエリアを使用して、フォームにクイックレポートを挿入することもできます (プラグインエリア参照)。

クイックレポートを作成するには、次の手順に従ってください:

1. **ツール**メニューから**クイックレポート...**を選択するか、ツールバーの対応するボタンをクリックします。
   4Dはクイックレポートエディタを表示します。以前にレポート定義が行われていた場合、その定義がウインドウ上に表示されます。ウインドウの内容を消去するには、**ファイル**メニューから**新規**を選択するか、このエディター標準のツールバーから**新規**アイコンをクリックして、新しいクイックレポートの設計を開始します。
2. クイックレポートエディターの左下隅にある**リスト**または**クロスタブ**ボタンをクリックし、生成するレポートのタイプを選択します。
   デフォルトではリストタイプが選択されています。選択したタイプに応じて、クイックレポートエリアが変わります。
3. レポートの定義を開始します。

### クイックレポートデザインの読み込みと保存

クイックレポートのデザインをファイルとして保存し、そのファイルをクイックレポートエディターで開くことができます。クイックレポートのデザインにはレポートに関するすべての仕様が納められますが、データは含まれません。レポートのデザインを保存することにより、クイックレポートデザインのライブラリを保持し、必要に応じてそれらを利用できるようになります。
クイックレポートエディターの使用中に、既存のデザインファイルを読み込むことで、同じレコードセレクションを使用したまま、さまざまなレポートを印刷することができます。また、同じクイックレポートデザインを繰り返し使用して、さまざまなセレクションのレコードを印刷することもできます。

**注:** 同一ワークセッション中にクイックレポートエディターを閉じて開くと、最後に作成したデザインが再表示されます。

クイックレポートデザインを保存するには、次の手順に従ってください:

1. **ファイル**メニューから**別名で保存...**を選択します。
   または
   ファイル管理用ツールバーの**別名で保存...**アイコンをクリックします。
   4Dにより、ファイル保存用のダイアログボックスが表示され、ここでク
2. クイックレポートのファイル名を入力し、**保存**ボタンをクリックします。
   4Dはレポートをクイックレポートエディターで開くことができるファイルとして保存します。このファイルは、ハードディスク上に保存することができます。保存後にレポートのパラメータを修正し、それを再度保存したい場合は、**ファイル**メニューの**保存**コマンドを選択します。ディスク上のファイルが新しいファイルで置き換えられますが、その際に標準のファイル保存用ダイアログボックスは表示されません。

レポートデザインを読み込むには、次の手順に従ってください:

1. **ファイル**メニューから**開く...**を選択します。
   または
   ファイル管理用ツールバーの**開く...**アイコンをクリックします。
   4Dにより、ファイルオープン用のダイアログボックスが表示され、利用可能なクイックレポートデザインのリストが示されます。
2. ファイル名をダブルクリックするか、またはリストからファイル名を選択して**開く**ボタンをクリックします。
   4Dは、表示中のデザインを開かれたもので置き換えます。

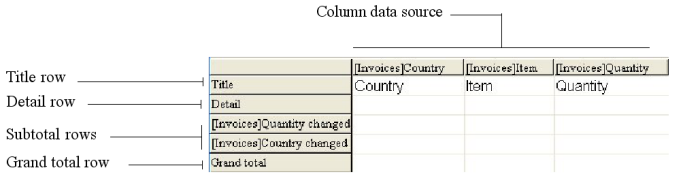
## クイックレポートエディター

### クイックレポートエディターの説明

クイックレポートエディターには、次の項目があります:

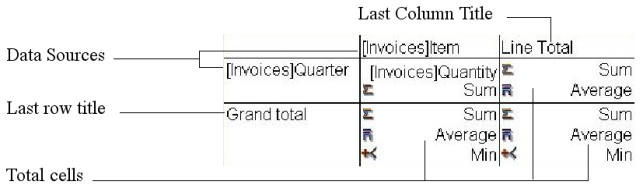
- **マスターテーブルドロップダウンリスト**: このメニューを使い、レポート生成の基盤として使用するマスターテーブルを指定します。このテーブルの各フィールドがフィールドリスト上に表示され、このテーブルに関連するリレートフィールドも表示されます。
- **フィールドリスト**: このリストから任意のフィールドを、ダブルクリックまたはドラッグ＆ドロップでレポートに挿入することができます。リストには、マスターテーブルのフィールドのリスト、テーブルとリレートフィールドの階層リスト、またはデータベースの全テーブルと全フィールドの階層リストのいずれかが表示されます。このエリアのすぐ上にあるフィールド選択リストから表示タイプを選択します。
  インデックス付きフィールドは太字で表示されます。また、リレートテーブルのフィールドを表示して選択することも可能です。
  **注:** クイックレポートエディターのフィールドリストには、非表示属性が設定されていないテーブルとフィールドだけが表示されます。テーブルやフィールドの非表示設定に関する説明は、テーブルプロパティおよび フィールドプロパティの"属性"項目を参照してください。
- **すべてのリレートを自動にする**: このオプションを使用し、クイックレポートエディタがデータベースの各種テーブル間のリレートを利用する方法を指定します。
  原則として、エディターでは自動リレートだけが利用可能です。デフォルトでは、このオプションは選択されておらず、自動リレートだけが使用されます。**すべてのリレートを自動にする**オプションを選択すると、クイックレポートエディタはデータベースの手動リレートを自動リレートとみなすため、データベースにリレートするすべてのデータへアクセスできるようになります。
  **開発者への注意**:
  - **すべてのリレートを自動にする**オプションが選択されると、クイックレポートエディターが閉じられた後も、カレントプロセスの間はすべてのリレートが自動に変更されたままになります。クイックレポートエディターの使用後、確実にすべてのリレートを手動にしたい場合は、次の命令を実行してください。
    ```
    SET AUTOMATIC RELATIONS(False;False)
    ```
  - QR REPORTコマンドを使用してエディターが呼び出されると、このオプションは表示されず、開発者はリレートの状態を直接管理できるようになります。
- **カラムディバイダ**: これらのラインは、レポートのカラム間の境界を示します。これらのラインを手動で移動し、各カラムのサイズを広げたり、または縮めることができます。**自動幅設定**オプションが選択されているカラムに対し、手動でサイズ変更を行うと、自動幅設定オプションが解除されます。
- **セル**: セルは、行とカラムが交差する部分です。
- **スクロールバー**: スクロールバーを使用してフォームエリアの表示領域を移動することで、クイックレポートデザインの任意の部分を同エリアに表示させることができます。
- **並び替え順リスト／並び替え順表示エリア**:
  - リストモードにおいて、このリストには並び替えを行うレポートの各フィールドとその並び替え順序、並び替え方向 (昇順または降順) が表示されます。このリストにフィールドを挿入するたびに、小計行がクイックレポートエリアへ追加されます。
  - クロスタブモードにおいて、このエリアには各データソースの並び替え順序が表示されます。
- **クイックレポートエリア**: このエリアを使用し、ドラッグ＆ドロップやダブルクリック、あるいはコンテキストメニューを用いて任意のフィールドを挿入し、レポートを作成することができます。また、追加されたカラムの幅やセルの色、罫線などを調整したり、フォーミュラや小計を削除するなど、様々な操作が可能です。

*リストモード*:




  - **タイトル行**: この行には、レポートに挿入されたフィールド名やフォーミュラが表示されます。タイトル行はレポートのページごとに繰り返されます。デフォルトとして、挿入されたフィールド名が表示されますが、その表示内容を変更することもできます。
  - **詳細行**: この行には各レコードから取り出された情報が納められ、レコードごとに詳細行がレポート上で繰り返されます。データタイプに応じて、表示フォーマットを割り当てることができます。
  - **小計行**: 小計行には、小計とそれに関連する文言が表示されます。各並び替え順に対して、この行が作成されます。
  - **カラムデータソース**: これらのタイトルは、各カラムのデータソースを示します。

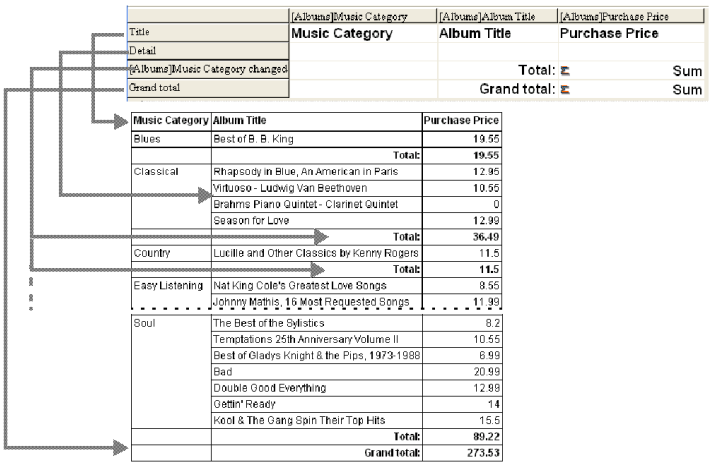
*クロステーブルモード*:



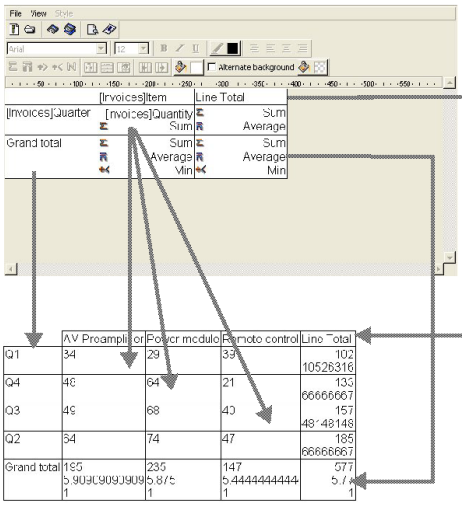

  - **データソース**: この2つのセルは、配列の2つのカテゴリーとして使用されるフィールドを格納します。また、中央のセルには、フィールドと計算式を同時に指定することができます。
  - **合計セル**: これらのセルは、そのカラム内容に関する計算結果を格納するために設けられています。
  - **タイトルセル**: これらのセルは、最終カラムまたは最終行のタイトルを格納します。この内容は変更可能です。
  - **コンテキストメニュー**: クイックレポートエディターにはコンテキストメニューがあり、これを使用することにより、特定の行やカラムへのアクセスやセルの操作を容易に行うことができます。メニューを選択したり、セルやカラムのプロパティエリアを用いて作業を行う代わりに、クイックレポートコンテキストメニューを表示して、特定の操作を実行することができます。

次の図は、完成したクイックレポートデザイン、およびデザイン仕様と印刷結果の関係を示しています:

**リストレポート:**




**クロステーブルレポート:**




**テキストの追加と修正**

クイックレポートフォームにおいて、テキストの追加や修正を行い、レポートの各部分にラベルを設定することができます。例えば、合計計算を実行する場合、小計行や総計行の他のセルにテキストを追加することにより、合計計算のラベルを設定することができます。次のように、テキストを追加、または修正することができます:

- 4Dによりレポートのタイトル行へ自動的に追加されるテキストを編集できます。
- 小計行と総計行の空セルにテキストを挿入できます。
- 小計フィールドの値を小計行に挿入できます。
- レポートに表示される任意のテキストに対して、フォント、フォントサイズ、行揃え、スタイルを指定できます。

レポートのセルにテキストを追加するには、クイックレポートフォーム上の空セルをダブルクリックします。そのセルにテキスト挿入ポイントが表示されます。集計計算のラベルを入力する場合は、集計アイコンを含むセルと同じ行にあるセルを選択します。ただし、集計結果を格納するセルにはテキストを入力することができません。

セルのテキストを修正するには、セルをダブルクリックしてカーソルを挿入ポイントに変え、修正しようとするセルのテキストをドラッグして選択します。選択されたテキストが反転表示されますので、新しいテキストを入力します。

**カラムの追加と挿入**

リストモードでは、フィールド名をフィールドリストからクイックレポートエリアにドラッグしてカラムを作成します。リレートテーブルのフィールドを追加したい場合には、フィールドの階層リストにある外部キーフィールドを展開し、リレートテーブルのフィールドを表示します (リスト上部にあるドロップダウンメニューからリレートテーブルまたはすべてのテーブルを選択している場合)。リレートテーブルのフィールドを追加するには、リレートが自動であることが必要です。手動のリレートも使用したい場合には、すべてのリレートを自動にするオプションを選択します。すると、エディターはデータベースの全リレートを自動に変更します。
クイックレポートには空のカラムを挿入することもできます。カラムの挿入後、そのカラムに対してフィールドやフォーミュラを割り当てることができます。

**注:** クロスタブモードでは、必要となるカラムをすべて使用してレポートが作成されるため、カラムの追加や編集は行えません。

フィールドの追加は、次のように行います:

1. クイックレポートエリアの既存カラムの右側へフィールド名をドラッグし、マウスボタンを放します。
   または
   フィールドリスト上のフィールド名をダブルクリックします。
   **注:** 既存カラムの上にフィールドをドラッグ＆ドロップすると、新しいカラムで置き換えられます。
   4Dは、そのフィールドのカラムを作成し、カラムヘッダとヘッダ行のセルの両方にフィールド名を配置します。
   デフォルトとして、4Dはクイックレポートの各ページの一番上にフィールド名をカラムヘッダとして印刷します。

メニューコマンドを使用してカラムを追加または挿入するには、次の手順に従ってください:

1. エディターの**カラム**メニューから**追加...**または**挿入...**コマンドを選択します。
   または
   クイックレポートエリアの未使用エリア上で右クリックし、**追加...**または**挿入...**コマンドを選択します。

4D標準のフォーミュラエディターが表示され、新しいカラムのデータソース (フィールドまたはフォーミュラ) を指定することができます。

**カラムの削除**

リストモードでクイックレポートにフィールドを指定するにつれて、カラムを移動したい場合があります。あるいは、レポートからカラムを削除したい場合もあるでしょう。

選択したカラムを削除するには、**カラム**メニューから**削除**を選択するか、削除したいカラムのヘッダ上で右クリックし、コンテキストメニューから**削除**コマンドを選択します。

4Dはレポートの選択されたカラムを削除します。

**カラムの置換**

リストモードでは、クイックレポートのカラム上に別のフィールドをドラッグすることにより、そのカラムを置き換えることができます。フィールドをフォーミュラで置き換えたり、またその逆も可能です。

カラムをフィールドで置き換えるには、次の手順に従ってください:

1. 置き換えようとするカラムへ、フィールドリストからフィールドをドラッグします。
   置き換えは即座に行なわれます。置換前のフィールドが並び替えレベルに指定されていた場合には、並び替え順リストにおいても、フィールドが置き換えられます。

を使用してカラムを置き換えるには、次の手順に従ってください:

1. 置き換えようとするカラムを選択し、**カラム**メニューから**編集...**を選択します。
   または
   置き換えられるカラムのヘッダ上で右クリックし、コンテキストメニューから**編集...**コマンドを選択します。
   4D標準のフォーミュラエディターが表示され、そのカラムの新しいデータソースを指定することができます (フィールド、またはフォーミュラ)。
   クイックレポートを印刷する際、4Dは詳細行に表示される各レコードごとに、フォーミュラの計算結果を印刷します。クイックレポートにフォーミュラを追加する方法についての詳細は、クイックレポートにフォーミュラを関連付けるを参照してください。

**カラムサイズの調整**

**自動幅設定**ボタンが示すように、デフォルトではクイックレポートエディターはカラムサイズを自動的に調節します。各カラムのサイズは、カラムに表示されるデータ、およびカラムに入力されたラベルの最大長に基づいて調整されます。クイックレポートエディターは、レポートの印刷時にカラムサイズを決定します。

カラムサイズの自動処理が可能になるのは、そのカラムに**自動幅設定**属性が設定されている場合です。このオプションの設定を行ったり、設定状況を調べるには、カラムメニュー、またはカラムのコンテキストメニュー、あるいはカラムツールバーの対応するボタンを使用して行います。

各カラムの幅を確認するには、画面上にレポートのプレビューを表示します。レポートのプレビューに関する詳細は、クイックレポートの生成を参照してください。

**自動幅設定**を選択すると、印刷されるレコードデータの最大幅に基づいてカラム幅が変更されるため、異なるレコードを選択するとカラムサイズが変わる可能性があります。

カラムサイズを手動で変更することもできますが、その場合**自動幅設定**オプションは自動的に解除されます (適用可能な場合)。カラムサイズが手動で設定されている場合には、カラム内のテキストは折り返し表示されます。

**カラムの移動**

リストモードでクイックレポートへフィールドを指定するにつれて、いくつかのカラムを移動して他の場所に配置したい場合があります。選択したカラムを移動するには、**カラム**ツールバーにある移動用のアイコン をクリックしてカラムを左右に移動します。あるいは、**カラム**メニューから**右へ移動**または**左へ移動**を選択し、カラムを希望する位置に移動します。

4Dは選択されたカラムを移動します。

## レポートのグラフィック属性を変更する

クイックレポートに関するグラフィック上の外観を変更することができます。クイックレポートエディターを使用して、次の属性を設定することができます:

- 文字のフォント、フォントサイズ、整列、スタイル、カラー
- セルの背景色
- セルの罫線

さらに、4Dが提供する一連の定義済レポートテンプレートを自由に修正して利用することができます。

**注:** 外観に関する設定は、出力先が"プリンタ"と"HTML ファイル" (カラーのみ) である場合にのみ考慮されます。これ以外のタイプの出力先の場合は、レポートを生成した後で、出力先であるアプリケーション (4DView、またはテキストエディター) を使用して、レポートのフォーマットを行なわなくてはなりません。レポートの出力先に関する詳細については、クイックレポートの生成を参照してください。

### 文字フォント、フォントサイズ、整列、スタイル、カラー属性の設定

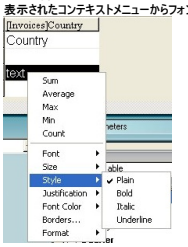
クイックレポートを設計する際、各種フォントやフォントサイズ、整列、スタイル、色を指定することができます。これらの設定は、クイックレポートの行、カラム、またはセル内のテキストやデータ、集計結果に適用することができます。

レポートの詳細行に対して設定を行った場合、レポートのプレビューや印刷を行うまで結果を検証することができません。

クイックレポートのメニューコマンドまたはクイックレポートのコンテキストメニューを使用して、フォント属性を指定することができます。

文字のフォント、フォントサイズ、スタイル、整列、色を指定するには、次の手順に従ってください:

1. フォント属性を適用したい行ラベル、カラムヘッダ、またはセル上で、右クリックします。
2. 表示されたコンテキストメニューからフォント、サイズ、スタイル、整列、文字色サブメニューを使用して、フォント属性を自由に変更します:




または

1. レポート上で、設定したいカラム、行、セルを選択します。
2. フォントドロップダウンリストからフォントを、サイズドロップダウンリストからフォントサイズを選択します。次に、スタイルツールバーからスタイルと整列、文字色を選択します:

**注:** スタイルと整列は**スタイルメ**ニューからも指定することができます。

4Dは、選択されたエリア内にあるテキストやデータ、集計結果に対してこれらの設定を適用します。

### セルの背景色を指定する

クイックレポートエディターを使用し、各セルの背景色を設定することができます。単一色、あるいは2種類の代替色のいずれかを設定可能です。背景色を設定すると、表がより読みやすくなります。

リストタイプとクロステーブルタイプの両レポートに対し、背景色を設定することができます。背景色は、出力先が"プリンタ"および"HTML"の場合に使用します。

背景色を設定するには、次の手順に従ってください:

1. セル、カラム、または行を選択し、ツールバーの背景色パレットから主となる色を選びます:
   行、セル、カラムに対して好きな色の組み合わせを指定することができます。
2. (任意) 代替背景色を使用したい場合には、代替背景色パレットからも色を選択します:
   2種類の代替背景色が設定されたセルは、クイックレポートエディター上で両方の色を表示します。

次の図は、代替背景色が設定されたレポートがリストモード (プレビュー) で生成された例を示しています:

| 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 | Sound 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 |

HTMLフォーマットで作成されたレポートの場合も、背景色が反映されます。

| | [Albums]Album Title | [Albums]Musician | [Albums]Purchase Price |
|---|---|---|---|
| Title | Album Title | Musician | Purchase Price |
| Detail | | | |
| Grand total | | | |

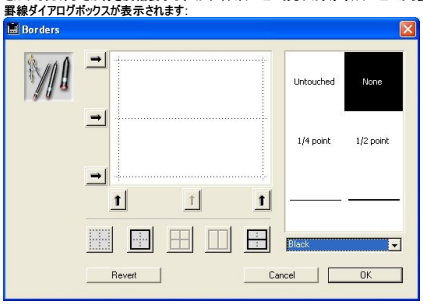
| Album Title | Musician | Purchase Price |
|---|---|---|
| Rhapsody in Blue, An American in Paris | George Gershwin | 12.95 |
| Sound of Jazz | Lionel Hampton | 12.2 |
| Nat King Cole's Greatest Love Songs | Nat King Cole | 8.55 |
| The Best of the Sylistics | Sylistics, The | 8.2 |
| Johnny Mathis, 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 |

### 罫線を設定する

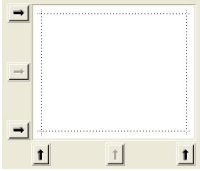
クロステーブルレポートとリストレポートの双方で、セルに対して罫線を設定することができます。

セル、カラム、または行の罫線を設定するには、次の手順に従ってください:

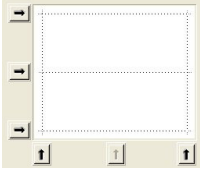
1. セルやカラム、または行を反転表示し、コンテキストメニューあるいは**スタイル**メニューから**罫線...**を選択します。
   罫線ダイアログボックスが表示されます:



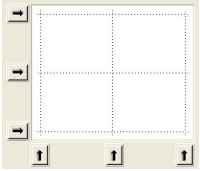

2. プレビューエリアは選択範囲のタイプに応じて動的に変わります。つまり、選択範囲がセル、カラム、または行であるか、さらにレポートがリストタイプ、またはクロステーブルタイプであるかによっても変わります。例えば、クロステーブルレポートの中央セルの特性は、リストレポートの詳細セルの特性とは異なります。クロステーブルレポートの中央セルは、横方向および縦方向に繰り返されますが、リストレポートの詳細セルは縦方向にのみ繰り返されます。また、その他のセルでは繰り返しが全く行われない場合もあります (例えば、タイトルなど)。
   - 選択したセルがタイトルセルの場合 (これは繰り返されません)、外側の線はそのセルの外枠を表わします:



   - 選択したセルがリストレポートの詳細セルの場合 (縦方向に繰り返されます)、外側の線はその繰り返されたセルの外枠を表わし、内側の横線は連続する2つのセル間の罫線を表わします:



   - 選択したセルがクロステーブルレポートの中央セルの場合 (横方向と縦方向に繰り返されます)、外側の線はその繰り返されたセルの外枠を表わし、内側の線は中央セルを元に生成された各セル間の罫線を表わします:



3. 使用する線の太さと色を選択します。
   各種罫線に対し、さまざまな色を使用することができます。
4. 矢印ボタンまたはショートカットボタンを使用して、罫線を選択します。
   編集しているセルのタイプに合わせて、いくつかのショートカットボタンが使用不可になっていることが分かります。
5. 設定する罫線ごとにステップ2とステップ3を繰り返し、**OK**をクリックして変更を有効にします。
   **元に戻す**をクリックすれば、元の罫線設定に戻せます。

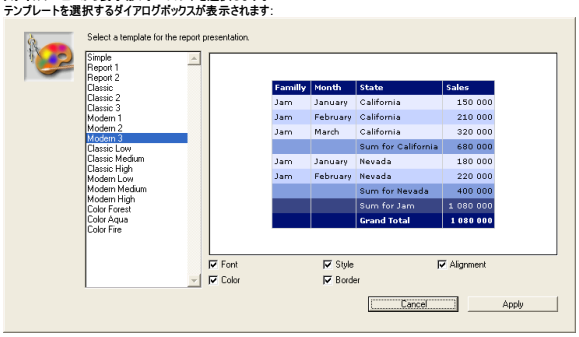
**定義済レポートテンプレートを使用する**

クイックレポートエディターには一連の定義済レポートテンプレートが用意されており、プリンタやHTML へのレポート作成の際にそれらを利用して、グラフィック上の各種設定を行うことができます。テンプレートはすべて、そのままで使用することも、あるいは独自のレポートを作成する際の基盤として利用することも可能です。テンプレートはいつでも適用することができます。

レポートテンプレートは、一連のグラフィック属性であるフォントやカラー、スタイル、罫線、行揃えを組み合わせたものです。テンプレートのすべての属性、あるいはその一部を利用することができます。

定義済テンプレートを使用するには、次の手順に従ってください:

1. **スタイル**メニューから**表示形式...**コマンドを選択します。
   テンプレートを選択するダイアログボックスが表示されます:




2. 定義済テンプレートリストにおいて、使用したいテンプレートをクリックします。
   すると、右側のエリアに各テンプレートのプレビューが表示されます。設定はいつでも修正できる点に留意しながら、目的に適したテンプレートを選択してください。
3. テンプレートで設定されているグラフィックオプションの一部だけを使用したい場合には、取り入れたくないオプションの選択を解除することができます。
   デフォルトでは、すべてのオプションが選択されています。例えば、罫線オプションの選択を解除すると、テンプレートで定義されている罫線はレポートへ適用されません。
   **注:** 出力先がHTML ファイルであるレポートの場合には、カラーオプションだけが考慮されます。
4. **適用**をクリックします。
   選択した属性は即座にレポートへ適用されます。事前にグラフィック属性が設定されていた場合には、それらが置き換えられます。属性のなかには、レポートが生成されて初めて表示されるものもある点に留意してください。

## レコードの並び替え

クイックレポートエディターの重要な機能の1つが、レポートのレコードの並び替えです。レコードを並べ替える理由としては、次の2つがあります:

- 一定の順序でレコードを表示する。
- グループごとの集計値を表示するために、レコードグループと小計エリアをレポートに作成する。

### リストレポートの並び替え順を指定する

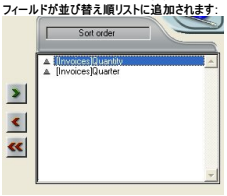
並び替え順は、並び替え順リストにカラムをドラッグ＆ドロップするだけで、随時に指定することができます。例えば、販売員レコードを販売地域フィールドで並び替えたい場合、販売地域カラムを並び替え順エリアにドラッグ＆ドロップします。

また、フォーミュラが格納されたカラムを選択して並び替え順リストに配置すると、そのフォーミュラに基づいた並び替えを行うことができます。クイックレポートへのフォーミュラ追加に関する詳細は、クイックレポートにフォーミュラを関連付けるを参照してください。

レポートは複数レベルで並べ替えることができます。並び替え順リスト上にフィールドやフォーミュラが表示される順番が、並び替えレベルを表わします。

フィールドリストを用いて並び替え順を指定するには、次の手順に従ってください:

1. 並び替えレベルとして設定したいフィールドをフィールドリストから並び替え順リストへドラッグします。
   または
   フィールドリストのフィールドを反転表示させ、フィールド挿入ボタン をクリックします。
   フィールドが並び替え順リストに追加されます:

   デフォルトとして、並べ替えは昇順に実行されます (A → Z)。並び替え順リスト上でフィールドの隣にある三角形をクリックすると、並び替え方向を逆に設定することができます。

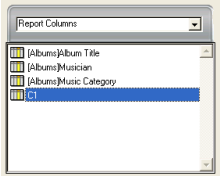
   **注:** 並べ替えられるフィールドがレポートカラムにまだ挿入されていない場合、クイックレポートエリア上でカラムが自動的に追加されます。

カラム (フィールドまたはフォーミュラ) の並び替えを指定するには、次の手順に従ってください:

1. テーブル選択ドロップダウンリストから、**レポートカラム**を選択します。

   すると、リストにはレポート上に存在する各カラムのヘッダが表示されます:

2. 並び替えレベルとして設定したいカラムをドラッグし、それを並び替え順リストにドロップします。
   または
   フィールドリストのフィールドを反転表示し、フィールド挿入ボタン をクリックします。
   すると、4Dは並び替え順リストにそのカラムの名前を表示します。

#### 並び替えレベルを変更する

並び替えのレベルを変更するには、次の手順に従ってください:

1. 並び替え順リストのフィールド上で右クリックして、コンテキストメニューを表示します。
2. **上**または**下**を選択し、並び替えレベルのフィールドを上位または下位に (ひとつずつ) 移動します。

#### フィールドやフォーミュラを並び替え順リストから削除する

並び替えリストから任意のフィールドやフォーミュラを削除することができます。

並び替え順リストからフィールドまたはフォーミュラを削除するには、次の手順に従ってください:

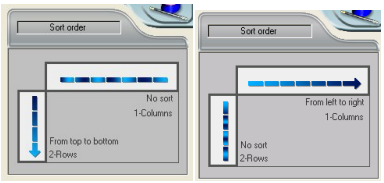
1. 並び替え順リストから削除しようとするフィールドまたはフォーミュラを選択し、右クリックしてコンテキストメニューを表示します。
2. **削除**を選択します。

すると、そのレベルが削除されます。関連するカラムはレポートから削除されない点に注意してください。反対に、関連する小計行は消去されます。

### クロステーブルレポートの値を並び替える

クロステーブルレポートでは、横方向と縦方向のデータソース (最終的な表でカテゴリーとして使用される2つのデータソース) の値だけを並び替えることができます。

クロステーブルレポートのカテゴリーを並び替えるには、並び替え順エリアで並び替えインジケータをクリックします。矢印は、指定された並び替え方向を示します:




データソースの並び替え順を変更、または削除するには、インジケータを再度クリックします。指定できる各種レポートが続けて表示されます。矢印が表示されない場合、並び替え順は選択されていません。この場合、値はセレクションにおける出現順に表示されます。

### 小計レベル

クイックレポートにおいてブレークレベルを設定し、1つ以上の並び替えフィールドの値に従って、レコードをグループ別に分類、つまり"ブレークさせる"ことができます。各ブレークレベルごとにブレークエリアが印刷されます。また、ブレークエリアには集計値を印刷することができます。集計計算 (合計、平均、最小値、最大値、カウント、標準偏差) は、各レコードグループ単位に算出されます。

ブレークレベルは、並び替えレベルとブレーク行により決定します。例えば、販売地域でレコードを並び替える場合、4Dは販売地域が同じである各レコードグループの間にブレークを挿入します。これらの行は、並び替えを指定すると自動的に挿入されます。

クイックレポートに小計行を追加した後は、各ブレーク単位で集計計算を実行させることができます。例えば、小計行に集計計算を挿入し、ある営業地域における各地区の売上小計を表示することができます。小計行と総計行に集計計算を追加する方法については、集計計算の追加を参照してください。

小計行のラベルは、ブレークが発生する値の変わり目を示します。

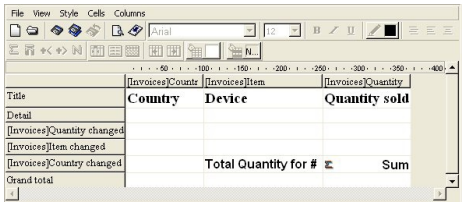
**ブレークフィールドの値をラベルに使用する**

ブレークフィールドの値を用いて各小計行にラベルを設定することにより、レポートの見栄えをよくしたり、より分かりやすくすることができます。

ブレークエリアに設定されたラベルにブレークフィールドの値を印刷するには、ラベル内で数値記号 (#) を使用します。例えば、"部門別給与合計: #"というテキストを指定すると、レポートの印刷時に数値記号の代わりに部門名 (この場合、部門フィールドの値) が挿入されます。

数値記号 (#) は、ブレークフィールド以外のカラムでも使用できます。数値記号 (#) により、小計行の任意のセルにブレークフィールドの値を表示することができます。

次の図は、小計行のラベルにおける数値記号の使用例を示します:



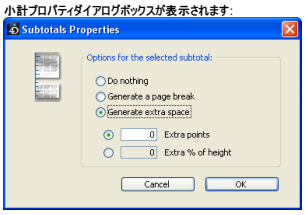

**小計の行間指定**

小計行の行間を設定することで、クイックレポートのページレイアウトと外見を変更できます。例えば、小計ごとに改ページを入れることが可能です。

小計のオプションを使って、レポートの各部分の区切りを際立たせることができます。

小計行の行間を設定するには、次の手順に従ってください:

1. 任意の小計行を選択し、エディターの**ファイル**メニューから**小計プロパティ...**をクリックします。
   または
   小計行の上で右クリックし、コンテキストメニューから**小計プロパティ...**を選択します。
   小計プロパティダイアログボックスが表示されます:




2. 次の3つのオプションから選択します。
   - **なにもしない** (デフォルト設定) : 小計行に対して特段の設定を行いません。レポート内の他の行と同じ行間設定が適用されます。
   - **ページブレークを生成する**: レポート内の各小計行の後に改ページが挿入されます。
   - **追加のスペースを生成する**: レポート内の各小計行の後に指定した高さのスペースを挿入します。
     スペースの高さを指定するには、次の方法があります:
     - **ポイント追加**: 追加スペースの高さをピクセル単位で指定します。
     - **% (高さ) 追加**: レポート内の通常行の高さを100%として追加スペースの高さを%で指定します。
       例えば、2行の空白行に相当するスペースを追加したい場合には、200という値を入力します。

# 集計計算の追加

### クイックレポートにフォーミュラを関連付ける

クイックレポートのカラムにフォーミュラを追加することができます。例えば、年間給与フィールドを基に従業員の月給を算出するフォーミュラを追加することができます。

カラムにフォーミュラを関連付けるには、次の手順に従ってください:

1. 空のカラムを挿入します。
   または
   既存のカラムをクリックして、**カラム**メニューから**編集...**を選択します。
   または
   既存のカラムヘッダをダブルクリックします。
   または
   既存のカラムヘッダ上で右クリックして、クイックレポートのコンテキストメニューを表示し、**編集...**を選択します。
   4Dによりフォーミュラエディターが表示され、ここでフォーミュラを作成することができます。既存のカラムを選択した場合、カラムの内容が作成したフォーミュラで置き換えられます。
   **注:** 作成したフォーミュラがカレントセレクションを変更しないことを確認してください。レポートはカレントセレクションに基づいて作成されるため、カレントセレクションが変更されると、クイックレポートの印刷時に問題が生じる可能性があります。カレントセレクションを変更するコマンドと関数については、ランゲージリファレンスマニュアルを参照してください。
2. フィールド、演算子、コマンド、あるいは4Dメソッドを選択してフォーミュラを作成し、希望する値を編集エリアに入力します。
   または
   **読み込み...**ボタンをクリックし、ディスクから既存のフォーミュラを読み込みます。
   フォーミュラエディターに関する詳細は、フォーミュラエディターを参照してください。
   他のカラムやレポートに読み込んで使用できるように、フォーミュラをファイルとして保存するには、**保存...**ボタンをクリックし、ダイアログボックスにファイル名を入力します。
3. **OK**ボタンをクリックして、カラムにフォーミュラを割り当てます。
   4Dは、フォーミュラであることを識別する新しいラベルをカラムに付加します。このラベルは、カラムのヘッダセルにラベルを入力して変更することができます。フォーミュラには"*C1*"から"*Cn*"までのラベルが付けられます。これらのラベルは、カラムの現在値が格納される変数の名前です。これらの変数は、他のフォーミュラで使用することができます。

### 集計計算の追加

フィールドやフォーミュラの内容に基づいた集計計算を各小計行や総計行に追加することができます。クロステーブルレポートでは中央の行に計算を挿入することができます。

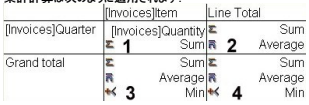
クイックレポートエディターのツールバーにある計算用ボタンは、クイックレポートで使用可能な集計計算オプションを示します:

次の集計計算を使用することができます:

- **合計**: レポートまたはブレーク内の値を合計する。
- **平均**: レポートまたはブレーク内の値の平均値を算出する。
- **最小**: レポートまたはブレーク内の最小値を表示する。
- **最大**: レポートまたはブレーク内の最大値を表示する。
- **カウント**: レポートまたはブレーク内のレコード数を算出する。
- **標準偏差**: レポートまたはブレークの平方偏差の平方根を表示する (平方偏差 (バリアンス) は平均前後でのばらつきの値です)。

また、これらのオプションは、小計行や総計行のセルのクイックレポートコンテキストメニューにも表示されます。クロステーブルレポートの場合は、合計セルと中央のセルで表示されます。

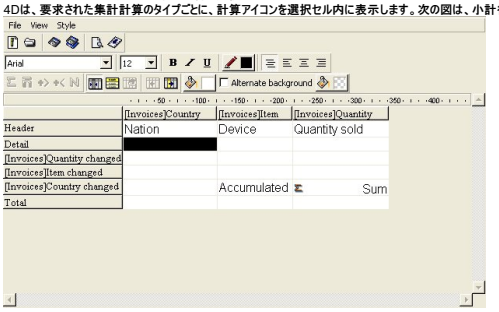
- **リストレポート**
  総計行に集計計算を指定すると、レポートのすべてのレコードを対象に計算が実行されます。小計行に集計計算を指定した場合は、各ブレーク内のレコードに対して別々に計算が行なわれます。
- **クロステーブルレポート**
  集計計算は次のように適用されます:




  **1**= テーブルの各セルに対して適用
  **2**= 各行の値に対して適用
  **3**= 各カラムの値に対して適用
  **4**= 最終カラムと最終行の両方の値に対して適用

集計計算を追加するには、次の手順に従ってください:

1. 集計計算を挿入しようとするセルを選択します。
   合計、最小値、最大値、平均値の計算は数値フィールドまたはフォーミュラに対してのみ有効です。
2. 必要な種類の集計計算用ボタンをクリックします。
   または
   コンテキストメニューから、任意の集計計算を選択します。
   4Dは、要求された集計計算のタイプごとに、計算アイコンを選択セル内に表示します。次の図は、小計行の合計計算アイコンを示しています:




#### 集計値とカラム値をラベルで使用する

次のコードを使用し、集計値をラベルに挿入することができます:

- "##S"は小計行または総計行の**合計**で置き換えられます。
- "##A"は**平均値**で置き換えられます。
- "##C"は**カウント**の値で置き換えられます。
- "##X"は**最大値**で置き換えられます。
- "##N"は**最小値**で置き換えられます。
- "##D"は**標準偏差**の値で置き換えられます。
- "##*xx*"の"*xx*"はカラム番号を表わし、そのカラムのフォーマットで表されたカラムの値で置き換えられます。このカラムが存在しない場合、置き換えは行われません。

これらのコードは、1つのセル内でラベルとデータを混在させたい場合に役立ちます。

#### ブレークカラムの同一値を表示する

小計を含むレポートにおいて、集計計算が行えるようにレコードをグループ化するために用いるカラムは、"ブレークカラム"と呼ばれます。次に示すレポートでは、レポート内のレコードが注文日、商品CD 毎にグループ化されているため、注文日、商品CDフィールドがブレークカラムとなります。

このようなレポートを印刷すると、ブレークカラムの値はブレークごとに1度だけ印刷されます。

つまり、注文日、商品CD は、そのグループの最初のレコードに対してのみ印刷され、注文日、商品CDが変わるまで繰り返し印刷されることはありません。

| 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 |

しかし、場合によっては、ブレークエリアの各レコードに対して表示されるように、ブレークカラムの値を繰り返し印刷したい時があります。これを行うには、カラムプロパティである**同一値も印刷**を選択します。このプロパティの選択は、ツールバーの**同一値も印刷**ボタンをクリックするか、そのカラムのクイックレポートコンテキストメニューから**同一値も印刷**メニューコマンドを選択、あるいは**カラム**メニューの**同一値も印刷**を選択して行います。

カラムのフィールドの値を繰り返し表示するには、次の手順に従ってください:

1. カラムのヘッダ行をクリックしてカラムを選択し、**カラム**メニューから**同一値も印刷**コマンドを選択するか、カラムツールバーの**同一値も印刷**ボタン
   をクリックします。
   または
   そのカラムのコンテキストメニューから**同一値も印刷**を選択します。このメニューでチェックマークが付いているプロパティがカラムに対して適用されます。
   レポートを印刷すると、ブレークカラムの値がレコードごとに繰り返し印字されます。

次の図は、前述のレポートの商品CDカラムに対して**同一値も印刷**を選択した場合の結果を示しています:

| First Name | Last Name | Department Name | Salary |
|---|---|---|---|
| Biff | Davis | Accounting | 43780 |
| Smeldorf | Garbando | Accounting | 19610 |
| Alan | Hull | Accounting | 41460 |
| Bryan | Pfaff | Accounting | 26440 |
| Shirley | Ransome | Accounting | 36040 |
| Marlys | Wilson | Accounting | 36500 |
| | | Sum for Department : Accounting | 203830 |
| Kathy | Forbes | Engineering | 18840 |
| Dennis | Hanson | Engineering | 40520 |
| Mary | Smith | Engineering | 55000 |
| Andy | Venable | Engineering | 43520 |
| Lance | Wolfram | Engineering | 27300 |
| | | Sum for Department : Engineering | 185180 |

## 表示フォーマットの設定

数値 (整数、倍長整数、実数)、文字、日付、時間、ピクチャータイプのデータを含む列に対して、表示フォーマットを指定することができます。

例えば、ある列に価格を表示する場合、価格フィールドの詳細セルに数値フォーマットを追加することができ。"¥####,##0;-¥####,##0"というフォーマットは、数値の左側に円記号 (¥) を配置し、-¥999,999から¥999,999までの金額を表示することができます。

また、レポートに電話番号や保険証番号のような文字フィールドが含まれる場合は、文字フォーマットを使用することができます。また、レポートで日付や時間、ピクチャーを表示する場合も、表示フォーマットを割り当てることができます。

4Dでは、さまざまなデフォルト表示フォーマットを提供しています。しかし、数値や文字フォーマット用に独自の表示スタイルを作成することもできます。この件に関する詳細は、表示フォーマットを参照してください。

数値、文字、日付、時間、ピクチャータイプのフィールドに表示フォーマットを割り当てるには、次の手順に従ってください:

1. フォーマットを割り当てたいカラムのセル上で、右クリックします。
   エディターのコンテキストメニューが表示されます。

2. **フォーマット**サブメニューから表示フォーマットを選択します。
   サブメニューの内容は、そのカラムに格納されたデータのタイプ (数値、文字、日付、時間、またはピクチャー) によって変わります。レポート列にフォーマットできないデータが納められている場合、コンテキストメニューには**フォーマット**コマンドが表示されません。

フォーマットは詳細セルに入力します。そのカラムに対して集計計算も設定している場合は、詳細セルに指定したフォーマットが自動的に集計値へも適用されます。ただし、"カウント"計算だけは例外で、常に整数型として表示され、円 (¥) 記号のようなフォーマット記号を受け入れません。
また、セルを二回クリックしてから、キーボードを用いてフォーマット要素を入力すれば、適用するフォーマットを手動で入力することもできます。

レポート内の列には、それぞれ異なるフォーマットを適用することができます。

## 行や列の表示と非表示

4Dでは、リストモードにおいて、クイックレポートの行や列を隠すことができます。隠された列や行は、必要に応じて再度表示することができます。レポートに集計値だけを納めたい場合などに、行を非表示にすると便利です。例えば、総計行と小計行の集計値だけを表示したい場合、詳細行を非表示にします。また、この機能を使用して、小計行や総計行を隠すこともできます。並び替え用にカラムを使用する必要があるが、その列をレポート上に表示したくない場合は、列を非表示にすることができます。

クイックレポートのコンテキストメニュー、列メニュー、または列ツールバーを使用して、行や列を隠したり、表示したりすることができます。

**注:** クロステーブルレポートでは、行や列を隠すことはできません。

行や列を隠すには、次の手順に従ってください:

1. 行や列のヘッダー上をクリックし、隠したい行や列を選択します。
2. 隠される行や列のヘッダー上で右クリックし、コンテキストメニューから**隠す**を選択します。
   または
   クイックレポートエディターの**列**メニューから**隠す**コマンドを選択します。
   または
   列ツールバーの ボタンをクリックします。
   4Dはそのカラムをグレー表示し、クイックレポートの印刷やプレビューの際にその行が表示されないことを示します。

| | [Invoices]Country | [Invoices]Item | [Invoices]Quantity |
|---|---|---|---|
| Title | Country | Item | Price |
| Detail | | ~~#### ### #####~~ | ~~$#### ####~~ |
| Grand total | | | Sum |

| | [Invoices]Country | [Invoices]Item | [Invoices]Quantity |
|---|---|---|---|
| Title | Country | Item | Price |
| Detail | | ~~#### ### #####~~ | $##### #### |
| Grand total | | | Sum |

### 隠された行やカラムを表示する

行やカラムが隠されている場合は、**カラム**メニューやコンテキストメニューの**隠す**コマンドの横にチェックマークが表示されます。さらに、カラムツールバーの対応するボタンが押された状態になります。

カラムメニューまたはコンテキストメニューから**隠す**コマンドを再び選択するか、カラムツールバーの対応するボタンをもう一度押すと、隠された行やカラムを表示することができます。これを行った後は、クイックレポートエリアに行やカラムが通常通りに表示されます。

## ページヘッダーとページフッターを追加する

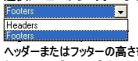
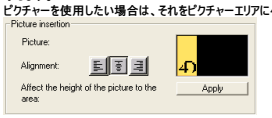
クイックレポートを印刷する前に、ページヘッダーページフッターを追加することができます。ページヘッダーとページフッターは、ヘッダー＆フッターダイアログボックスで指定します。

このダイアログボックスを使用し、次の操作を行うことができます:

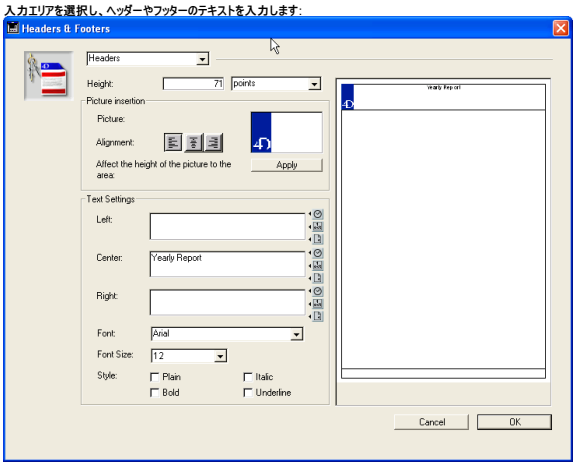
- ページヘッダーとページフッターのテキストやピクチャーを追加する。
- ページヘッダーエリアとページフッターエリアのサイズを指定する。
- ヘッダーとフッターの左側、中央、右側に個別のテキストを使用する。
- ページヘッダーとページフッターのテキストのフォント、フォントサイズ、スタイルを指定する。
- レポートにページ番号や日付、時刻を追加するコードを挿入する。

ページのヘッダーとフッターは、プリンタへ印刷する場合にのみ指定することができます。ただし、一度指定を行うと、出力先が変更されても、その設定がレポートに残されます。その他の出力先に関する詳細は、クイックレポートの生成を参照してください。

ページヘッダーとページフッターを追加するには、次の手順に従ってください:

1. **ファイル**メニューから**ヘッダー＆フッター...**を選択します。
   ヘッダー＆フッターダイアログボックスが表示されます。
   **注:** プレビューエリアでは、プリントフォーマットの設定が考慮されます。
   ヘッダー＆フッターダイアログボックスを使用すると、一つのウインドウでヘッダーとフッターの両方を指定することができます。ヘッダー/フッタードロップダウンリストを用いて、これから定義しようとするヘッダーまたはフッターのいずれかを指定します。
2. 選択メニューから**ヘッダー**または**フッター**を選択します:
   
   
3. ヘッダーまたはフッターの高さを高さエリアに入力します。
   初めてこのダイアログボックスに入力する時は、ヘッダーとフッターの高さがそれぞれ25ポイントに設定されています。これらの値は変更可能であり、さらにインチやセンチメートル単位で値を入力するために、その単位を変更することもできます。
   ヘッダーとフッターの高さを入力すると、ページプレビューエリアのラインの位置が変わり、印刷後のレポートに表示されるヘッダーとフッターのサイズを示します。
4. ピクチャーを使用したい場合は、それをピクチャーエリアにペーストします (通常のコピー＆ペーストコマンドを使用):
   
   
   - ピクチャーはヘッダーまたはフッターの中央に配置された状態でペーストされます。配置を変更するには、整列用のアイコンをクリックします:

   - ピクチャーの高さをヘッダー／フッターの高さとしたい場合、**適用**ボタンをクリックします:

5. 入力エリアを選択し、ヘッダーやフッターのテキストを入力します:
   
   

   各入力エリアの右側には3つのボタンがあり、入力エリアに変数を入力することができます。現在の実行時刻、実行日付、ページ番号を挿入することができます:

   また、変数用のコードを直接使用することもできます:
   - "#H": 実行時刻
   - "#D": 実行日付
   - "#P": 現在のページ番号
6. ヘッダーやフッターのフォント属性を指定し、**OK**をクリックして変更を有効にします。

## クイックレポートの生成

レポートの設計が完成したら、そのクイックレポートを"印刷"することができます。クイックレポートはさまざまな出力タイプに印刷することができます:

- 選択されている標準のプリンター
- ディスク上のテキストファイル
- 4DView ドキュメント
- HTML ドキュメント

出力先を選択するには、**ファイル**メニューから**出力先**を選択します。

### プリンター

このオプションは、現在の印刷設定で選択されているプリンターを使用します。プリンターに印刷する場合、印刷の前にレポートをプレビューすることができます。

プリンターに印刷するには、次の手順に従ってください:

1. **ファイル**メニューの**出力先**サブメニューから**プリンター**を選択します。
   デフォルトとして、このオプションが選択されています。
2. 必要に応じて、**ファイル**メニューから**用紙設定...**を選択し、印刷用のパラメータを指定します。
   **注:** このコマンドは、出力先として**プリンター**が選択されている場合にのみ使用可能です。
3. レポートのプレビューを行いたい場合は、**ファイル**メニューから**プレビュー...**を選択します。
   **注:** このコマンドは、出力先として**プリンター**が選択されている場合にのみ使用可能です。
   プリントプレビューダイアログボックスが現われ、定義したレポートがデータベースのカレントデータを用いて表示されます。
4. **ファイル**メニューから**出力...**を選択します。
5. 定義したレポートに適した設定を選択し、**OK**ボタンをクリックします。

### ディスクファイル

このオプションは、テキストエディターやスプレッドシートのような他のアプリケーションで開いたり、修正することができるディスクファイルにクイックレポートを送信します。また、このオプションは、クイックレポートのレコードをテキストファイルに書き出します。

このオプションを使用すると、4Dは自動的にカラムヘッダを最初の"レコード"として書き出します。

テキストファイルにレポートを生成するには、次の手順に従ってください:

1. **ファイル**メニューの**出力先**サブメニューから**ディスクファイル**を選択します。
2. **ファイル**メニューから**出力...**を選択します。
   4Dは標準のファイル作成用ダイアログボックスを表示し、ファイル名の入力を要求します。
3. ファイル名を入力し、**OK**ボタンをクリックします。
   4Dは、処理の進捗状況を知らせるダイアログボックスを表示します。ファイルへの書き出しが完了すると、4Dはクイックレポートエディターに戻ります。再び標準プリンターへクイックレポートを印刷したい場合は、必ず出力先を変更し直してください。

### 4DView ドキュメントへの印刷

このオプションは、4Dのデータ表示用プラグインである4DViewへレポートを送信します。**出力先**サブメニューから**4DView**項目を選択すると、レポートは表形式ではなく4DView ウインドウとして表示されます。4DView はレポートのすべてのデータを使用します。

**注: 4DView**という出力先は、お使いの4D環境にこのプラグインが既にインストールされており、適切なライセンスをお持ちの場合にのみ使用可能です。

4DView ドキュメントにレポートを生成するには、次の手順に従ってください:

1. **ファイル**メニューの**出力先**サブメニューから**4DView**を選択します。
2. **ファイル**メニューから**出力...**を選択します。

4Dはデータを格納した4DView ウインドウを作成します。4DView のメニューコマンドを使用して、その内容の変更や印刷、保存を行うことができます。

### HTML ドキュメントへの印刷

このオプションは、HTML ファイルへレポートを送信します。このオプションを選択すると、プログラムから変更を行わないかぎり、デフォルトのHTML テンプレートが使用されます。

このオプションを使用すると、4Dは自動的にカラムヘッダを最初の"レコード"として書き出します。

HTMLファイルへ印刷するには、次の手順に従ってください:

1. **ファイル**メニューの**出力先**サブメニューから **HTML ファイル**を選択します。
2. **ファイル**メニューから**出力...**を選択します。
   4Dは標準のファイル作成用ダイアログボックスを表示し、ファイル名の入力を要求します。
3. ファイル名を入力し、**OK**ボタンをクリックします。
   4Dは処理の進捗状況を知らせるダイアログボックスを表示します。HTML ファイルへの書き出しが完了すると、4Dはクイックレポートエディターへ戻ります。再び標準プリンターへクイックレポートを印刷したい場合は、必ず出力先を変更してください。

HTMLコードを生成する際には、データベース設定にて指定されたスタンダードセットが文字セットとして使用されます。(Web/オプション(II)ページ参照)

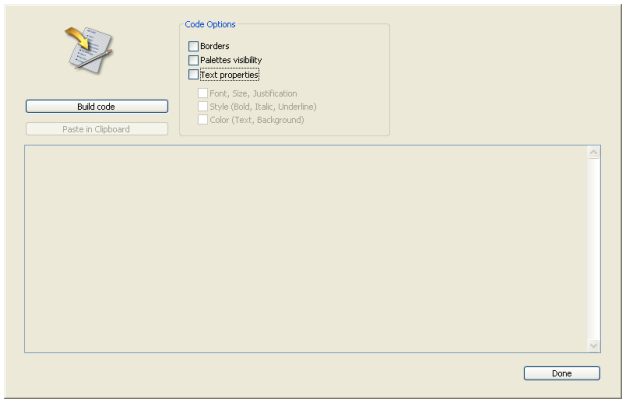
## 4D コードを生成する

クイックレポートエディターでは、定義したレポートに対応する4Dのコードを自動生成して、それをクリップボードにペーストすることができます。この機能により、4Dランゲージ (クイックレポート テーマのコマンド) を用いてカスタマイズできるクイックレポートのアウトラインを手動で作成することができます。この機能はデザインモードでのみ使用可能です。

コード生成を実行するためのボタンは、クイックレポートウィザードの最後のページにあります:

手動でクイックレポートを作成する場合でも、ウィザードの仕上げページを表示して、この機能を利用することができます。

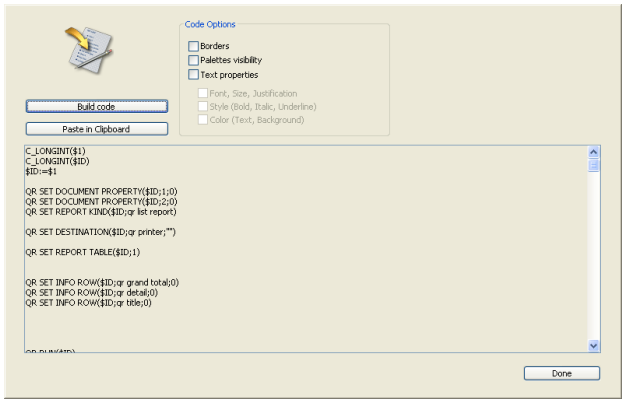
このボタンをクリックするとダイアログボックスが表示され、コード生成に関するパラメータを設定することができます:




次のオプションを利用できます:

- **境界線**: このオプションを選択すると、セル境界線のプロパティが生成された4Dコードに組み込まれます。
- **パレット表示**: このオプションを選択すると、カレントパレットの表示ステータス (標準、スタイル、カラムなど) が生成された4Dコードに組み込まれます。
  オフスクリーンエリアでこのコードを使用する予定であれば、このオプションは選択しないことをお薦めします。
- **テキスト設定**: このオプションを選択すると、レポートのフォントスタイル定義 (フォント、スタイルなど) が4Dコードに組み込まれます。このオプションを選択しない場合は、次の3つのオプションが無効になります:
  - **フォント、サイズ、行揃え**: このオプションを選択すると、レポートのフォントやフォントサイズ、行揃えに関する情報が生成後の4Dコードに組み込まれます。
  - **書体** (太字、斜体、下線) : このオプションを選択すると、レポートのテキストスタイルに関する情報が生成後の4Dコードに組み込まれます。
  - **色** (文字、背景) : このオプションを選択すると、レポートのテキストカラーに関する情報が生成後の4Dコードに組み込まれます。

オプションの定義が終了したら、**コードをビルド**ボタンをクリックして4Dコードを生成します。すると、ダイアログボックスのプレビューエリアにそのコードが表示されます:




コード用のオプションを変更して**コードをビルド**ボタンを再度クリックすると、プレビューエリアを更新することができます。

生成されたコードに不具合がなければ、**クリップボードへペースト**ボタンをクリックし、次に**終了**ボタンをクリックしてダイアログボックスを閉じます。

この後、標準のショートカットキーである**Ctrl キー+"V"** (Windows) や**Command キー+"V"** (Mac OS)、または**編集**メニューの**ペースト**コマンドを使用して、メソッドやテキストファイルにコードをペーストすることができます。

## ラベル

ラベルエディター
ラベルデザインを作成する
ラベルシートの作成

# ラベルエディター

4Dのラベルエディターにより、さまざまなラベルを容易に印刷することができます。ラベルエディターを使用し、次の操作を行うことができます:

- 宛名やファイルフォルダー、ファイルカードなど、その他数々のニーズに対応するラベルをデザインする。
- 装飾用のオブジェクトを作成し、ラベルテンプレートに挿入する。
- ラベルに使用するフォント、フォントサイズ、書体を指定する。
- 各ページの縦と横に印刷するラベル数を指定する。
- 各レコードにつき印刷するラベル数を指定する。
- ラベルページのマージンを指定する。
- 各ラベルまたは各レコードを印刷する際に実行するメソッドを指定する。
- ラベルデザインの読み込みと保存を行う。
- ラベルを印刷する。

ラベルは、フォームエディターを使用して作成することもできます。変数を含むラベルや、フォームエディターの描画用ツールを利用する特殊なラベルを作成する場合には、フォームエディターを使用してください。フォームエディターを用いたラベル作成についての詳細は、ラベルの作成を参照してください。

ラベルエディターを使用して、ラベルの作成やフォーマット、印刷を行います。ラベルエディターには、ラベルを設計したり、ラベル用紙上でのラベル位置を決めるための各種設定が用意されています。例えば、宛名ラベルを作成する場合、1行目に相手先の名字と名前、2行目にその人の住所を納めたラベルを設計することができます。ラベルエディターでは、ページ上のラベル数を指定したり、ラベル用紙のマージンを指定することで、ラベルテキストがラベルの中央に配置されるようデザインを調整することができます。
ラベルデザインが完成したら、いつでも再利用できるようにディスクに保存することができます。

ラベルエディターは、次のように開きます:

- デザインモードでは、**ツール**メニュー、あるいはツールバーにある**ツール**ボタンのメニューから**ラベル...**を選択します。
  または
- アプリケーションモードでは、PRINT LABELコマンドを使用します。

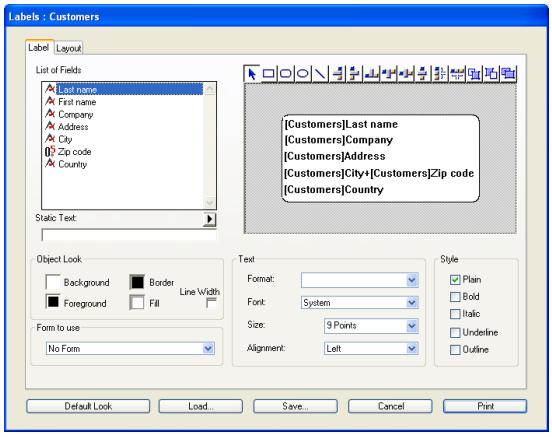
ラベルエディターは、ラベルページとレイアウトページという2種類のページから構成され、それぞれがダイアログボックスの上部にあるタブにより分けられています。ラベルページはラベルの内容を指定するために使用し、レイアウトページはページ上のラベルサイズや位置を指定するために使用します。

- **ラベル**ページには、ラベルのデザインやフォーマットを行うための設定があります。また、ラベルの作成に用いるフォームを指定することができます。
- **レイアウト**ページには、ラベルの用紙上の配置と印刷のための各種設定が用意されています。設定は手動で行うこともできますが、配置可能なラベル数をプログラムに計算させることもできます。ラベルのページにメソッドを割り当てることも可能です。

## ラベルページ

ラベルページには、ラベルのデザインやフォーマットを行うための設定があります。

ラベルページには、次の項目があります:




- **フィールドリスト**: このエリアには、カレントテーブルのフィールド名が階層リスト形式で表示されます。そのテーブルが別のテーブルにリレートしている場合は、外部キーフィールドにプラス記号 (Windows) または矢印 (Macintosh) が表示されます。リレートフィールドを展開すると、リレートテーブルのフィールドを表示することができます。リレートテーブルのフィールドはインデント (字下げ) されます。
  リストから右のラベルプレビューエリアにドラッグすると、フィールドをラベルテンプレートに追加できます。
  **注:** ラベルエディターには、非表示属性が設定されていないテーブルとフィールドだけが表示されます。テーブルやフィールドを非表示に設定する方法については、テーブルプロパティおよびフィールドプロパティを参照してください。

- **スタティックテキスト入力エリア**: このエリアを使用し、スタティックテキストオブジェクトをラベルに追加することができます。エリアに入力したテキストをプレビューエリアに追加するには、上の矢印ボタンをクリックします。ラベル上の配置はマウスを使って変更できます。

- **ラベルプレビューエリア**: このエリアを使用して、ラベルに表示させるすべてのオブジェクトの配置を調整し、ラベルを設計します。ラベルプレビューエリアには、選択したラベルのおおよそのサイズと形が表示されます (ラベルの大きさはレイアウトページを指定できます)。ラベルにはフィールドをドラッグしたり、スタティックテキストを挿入することができます。また、配置済みのフィールドの上にもう一つフィールドをドロップするだけで、二つのフィールド (例えば、名字と名前) を連結することが可能です。連結されたフィールドの狭間には、4Dのリソースに定義されたセパレータが挟まれます。デフォルトではスペースがセパレータに使用されます。
  エリア上部のツールバーには、オブジェクトの選択、描画、整列、配置、レイヤ操作、複製を行うためのツールがあります。また、クリップボードのオブジェクトをエリアへペーストすることも可能です。

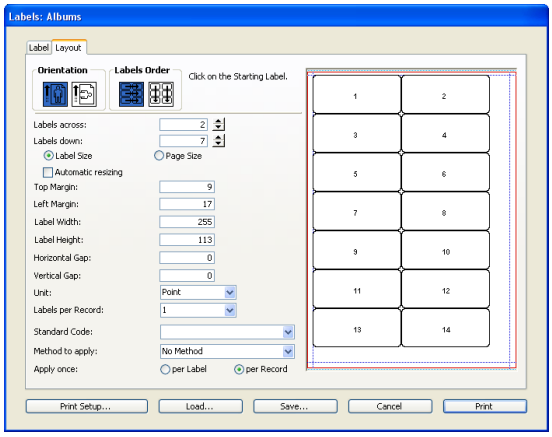
- **オブジェクトルック、テキスト、書体エリア**: これら三つのエリアでは、プレビューエリアに設置したアイテムについての設定を行います。オブジェクトルックの設定を使用して、ラベル上にある個々のオブジェクトの描画色と背景色、塗りつぶしパターン、境界線を指定することができます。テキストおよび書体の設定を使用して、テキストのフォントやフォントサイズ、表示フォーマット、書体などを指定することができます。これらのパラメーターはフィールドやスタティックテキストに対して適用できます。

- **利用するフォームドロップダウンリスト**: このドロップダウンリストを使用すると、ラベルエディターを利用せずに、フォームを用いてラベルを印刷することができます。ラベルエディターを使用してラベルを作成する場合は、このリストから**フォームなし** (デフォルト設定) を選択してください。フォームを使用したい場合は、このリストからそのフォームを選択します。すると、4Dはラベルエディターのラベル仕様をすべて無視し、指定されたフォームのデザインに従ってラベルを印刷します。すべての印刷ジョブと同様に、そのフォームに関連付けられたフォームメソッドやオブジェクトメソッドが実行されます。ラベル用フォームの設計に関する詳細は、ラベルエディターを使用するを参照してください。

- **デフォルトルックボタン**: このボタンは、選択されたオブジェクトにデフォルトのオブジェクトルック属性を適用します。オブジェクトルックエリアのパラメーターはすべて初期値にリセットされます。

## レイアウトページ

レイアウトページには、プリントマネージャー (Macintosh ではセレクタ) で選択したプリンターの印刷条件に基づいて、ラベル印刷用の各種設定が用意されています。これにより、ページを的確にフォーマットすることができます。




- **プリント設定...**ボタン: このボタンを使用して、ラベルを印刷するページのフォーマットを設定することができます。
  **注:** ラベルエディターで作成されるラベルシートは指定プリンターにおける印刷可能領域に基づきます (印刷に使用できないプリンターの強制マージンを除いた領域です)。プリンターにより指定されているマージンは、プレビューエリア内では青い線で表示されます。

- **方向とラベル順序** ボタン: これらのボタンを使用し、用紙方向、およびデータをラベルに割り当てる順序を指定することができます。ラベルシートの方向は用紙方向に左右されないという点に留意してください。このパラメータを変更する場合、それに応じてプリント設定ダイアログボックスでページの設定を必ず変更してください。

- **ラベル列数とラベル行数** ボックス: これらのボックスを用いて、ラベル用紙に表示されるラベル数を指定し、ラベルのサイズを制御します。

- **ラベルサイズとページサイズ**ラジオボタン: ラベルやページのサイズを設定するために、これらのボタンを用いてラベルまたはページを選択します。

  - ラベルサイズをクリックすると、対応するエリアにラベルの高さと幅を入力することができます。
  - ページサイズをクリックすると、右マージンと下マージンの値を入力することができます。これらのマージンはプレビューエリアにおいて青い線で表示されます。

- **自動サイズ調整**チェックボックス: 自動サイズ調整を選択すると、ラベル幅およびラベル高さ入力エリア内の値が自動的に設定され、他のパラメーターを変更する度に値が再計算されます。

- **マージン**ボックス: 選択されたラジオボタンに応じて、これらのボックスを使用してラベルのサイズやページサイズを指定できます。ラベル用紙のマージンを入力した後、ラベルテキストがラベルの中央に配置されるように、調整がさらに必要となるかもしれません。各マージンボックスに正数や負数を指定し、マージンを大きくしたり、小さくすることができます。

- **水平方向間隔**ボックス: このエリアは、ラベルの列間のスペースを制御します。

- **垂直方向間隔**ボックス: このエリアは、ラベルの行間のスペースを制御します。

- **単位** ドロップダウンリスト: このドロップダウンリストを使用すると、ラベルとラベルページのサイズ指定に使用する単位を変更することができます。ピクセル、ミリメートル、センチメートル、またはインチを使用することができます。

- **ラベル数／レコード**ドロップダウンリスト: このドロップダウンリストにより、レコード毎に印刷するラベルの枚数を指定することができます。印刷枚数を複数に設定した場合、ラベルはレコード毎に連続して印刷され、一つのレコードが指定回数印刷されてから次のレコードのラベル印刷に移ります。

- **ラベルテンプレート**ドロップダウンリスト: この項目では、ドロップダウンリストから一般的な市販のラベル用紙を選択することにより、ラベル、ページサイズ、マージンを指定することができます。

- **適用するメソッド**ドロップダウンリスト: このドロップダウンリストを使用して、印刷時に実行されるメソッドを選択することができます。例えば、各ラベルが印刷された日時を印字するメソッドを実行することができます。

- **実行タイミング**ラジオボタン: これらのラジオボタンを使用し、メソッドの実行をラベル毎またはレコード毎のいずれに行うかを指定します。この項目は、各ラベルを複数印刷し、かつ印刷時にメソッドを実行する場合にのみ有効です。

- **レイアウトプレビューエリア**: このエリアには、ラベルエディターで入力したサイズに基づき、ラベルページ全体の状態が縮小表示されます。また、ページのプレビューでは、用紙設定ダイアログボックスで選択した用紙サイズが反映されます。最初に印刷されるラベルを選択することも可能です。赤い線は実際のページサイズを示し、青い線は印刷可能領域のサイズを示します。

**ラベルデザインの保存と読み込み**

4Dを使用して、各ラベルデザインをファイルとして保存し、それをラベルエディターから開くことができます。ラベルデザインを保存すると、ニーズに合わせて利用できるラベルライブラリを設けることができます。
ラベルデザインには、ラベルページやレイアウトページで設定されたパラメーターが保存されます。

- ラベルデザインを保存するには、**保存...**ボタンをクリックし、保存するラベルデザインのファイル名とファイル場所を指定します。
- ラベルデザインを読み込むには、**読み込み...**ボタンをクリックし、ファイルオープン用のダイアログボックスで、読み込むラベルデザインのファイル名を指定します (4Dは、選択されたデザインで現在のラベルデザインを置き換えます)。

Windows 上のラベルファイルには".4LB"という拡張子が付加されます。

## ラベルデザインを作成する

ここではラベルエディターによるラベルデザイン機能を紹介します。各機能の説明については、前章を参照ください。ラベルエディターでは1つ1つのラベルのデザインと、ラベルシート全体のデザインを別々に、そして優先順位に沿った任意の順番で作成できます:

- ラベルが特定サイズのものであれば、まず**レイアウト**ページでラベルのサイズを指定してから、各ラベルのデザインを作成できます。
- ラベルが特定デザインのものであれば、まず**ラベル**ページでデザインを指定してから、ラベルシートのレイアウトをラベルに合わせて作成できます。

**注:** すべてのデザイン機能を有効にするには、**ラベル**ページにて利用する**フォーム**ドロップダウンリストから**フォームなし**オプションを選択している必要があります (このオプションの詳細ついては、ラベルページを参照ください)。

ラベルプレビューエリアには、選択したラベルのおおよそのサイズと形が表示されます。ラベルに項目を追加する前に、**レイアウト**ページでラベル用紙やラベルの大きさを指定することもできます。詳細については、ラベルシートの作成を参照してください。

### フィールドを挿入する

ラベルにフィールドを挿入するには、次の手順に従ってください:

1. ラベルページを開きます。
2. ラベル上に表示したい1番目のフィールドをフィールドリストからラベルプレビューエリアにドラッグします。
   リレートテーブルのフィールドを挿入したい場合は、リレートフィールドを展開してリレートテーブルのフィールドを表示します。フィールドがラベルプレビューエリアに追加されます。選択ハンドルは、そのフィールドが選択されていることを示しています。
3. 他にも挿入したいフィールドがあれば、続けて操作します。

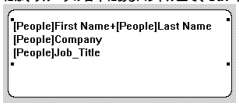
#### フィールドを連結する

ラベルをデザインする際に、複数のフィールド同士を連結させたい場合があります。例えば、名字と名前フィールドを連結して、ラベルの1行目に表示させたい場合です。また、フィールドを縦に連結することも可能です。

フィールドを連結することの利点は、フィールド同士の間隔が4Dにより自動的に管理されることです。

複数のフィールドを連結するには、次の手順に従って下さい:

1. 1つ目のフィールドをラベルプレビューエリアにドラッグします。
2. 2つ目のフィールドを既存のフィールドの上にドロップします。
   または
   **Shift**キーを押しながら新しいフィールドを既存のフィールドの上にドロップします。
   - ドラッグ & ドロップだけで2つのフィールドを連結すると、4Dのリソースに定義されたセパレータが使用されます。デフォルトではスペースが使用されます。フィールドオブジェクトのファイル名の間にある"+"は、各フィールドが1つの行内で連結されることを示します。4Dがラベルを印刷すると、同じ行内にある各連結フィールドの間にスペースが挿入されます。
   - **Shift**キーを押しながら2番目のフィールドをドラッグ&ドロップして、2つのフィールドを連結すると、改行文字がセパレータとして使用されます。ラベルの印刷時に2番目のフィールドが空である場合は、そのフィールドの場所が削除され、次のフィールドで置き換えられます。フィールドの内容が改行を含むテキストである場合には、その改行も考慮されます。
     この機能を使用例として、フィールドが空の場合でも空白行を生成せずに、複数の連結フィールド ([顧客]住所1、[顧客]住所2、など) を用いた住所ラベルを作成することができます。
     一連の連結フィールドに含まれる行数と、ラベルプレビュー上の表示が一致するように、サイズを自動調整することができます。これを行うには、グループの右下にあるハンドル上で、**Ctrl**+クリック (Windows) または **Command**+クリック (Macintosh) します。




連結されたフィールドを削除するには、次の手順に従って下さい:

1. 連結フィールドを含む行を選択します。
2. **Backspace** (Macintosh では**Delete**) キーを押します。
   行内で最後に連結されているフィールドが削除されます。

### グラフィックを追加する

ラベルエディターでは、ラベルのデザインにグラフィックを追加して、ラベルの見た目を向上させたり、レコード毎の有益な情報を追加することができます。ラベルには3種類のグラフィックを追加することができます:

- スタティックテキスト
- エディターで描画したグラフィックオブジェクト
- クリップボードにコピーした画像

#### スタティックテキストを追加する

ラベルデザインにスタティックテキストを追加して、フィールドタイトルや見出しなどを表示させることができます:

1. **スタティックテキスト**エリアにテキストを入力またはペーストします。
2. 矢印上ボタンをクリックします。スタティックテキストオブジェクトがラベルプレビューエリアの左下に追加されます。
3. テキスト項目をラベルに追加した後は、ドラッグしたり、または他のオブジェクトに揃えることにより、位置の調整を行えます。

#### グラフィックオブジェクトを追加する

ラベルプレビューエリアには、図形描画用のツールバーが設置されています。例えば、ラベルの外枠を作成することができます。

描画可能な図形は長方形、楕円形、直線などです。

グラフィックオブジェクトを描画するには、プレビューエリアの上にある描画ツールをクリックして選び、描画します。
正円や正方形、垂直線または水平線を描画したい場合には、**Shift**キーを押しながら描画することで、規則正しい形のオブジェクトを描けます。

**注:** ラベルに追加したフィールドの周りに境界線を追加するには、次のショートカットが使えます:
フィールドを選択し、**Ctrl**キー (Macintosh では**Command**キー) を押しながら、数値キーの**1**を押します。すると、フィールドから1ピクセルの距離を置いて境界線が描画されます。数値キーの**2**を押すと、2ピクセルの距離を置いた境界線が描画でき、9ピクセルまでこのショートカットで描画可能です。

#### 画像を追加する

例えばロゴマークなど、任意の画像をクリップボードからラベルにペーストすることができます。まずプレビューエリアを選択したら、あとは標準のショートカットキー**Ctrl+V** (Macintosh 上では**Command+V**) を使うだけです。

#### オブジェクトを複製する

ツールバーのボタンをクリックすると、選択したフィールドやグラフィックオブジェクトが複製できます。

### オブジェクトを選択する

1つ以上のオブジェクトに対して変更を同時適用するには、まずオブジェクトを選択する必要があります。フィールドやグラフィックオブジェクトを選択するには、いくつかの方法があります。

- 離れて配置されている複数のオブジェクトを選択するには、まず1つ目のオブジェクトをクリックして選択し、Shift キーを押しながら選択に追加したい他のオブジェクトをクリックします。
- 隣接しているオブジェクト群を選択するには、プレビューエリアをクリックし、そのまま長方形を描くようにマウスカーソルをスライドさせます。この長方形に接触するオブジェクトはすべて選択されます。
  この時、**Alt**キー (Macintosh 上では**Option**キー) を押しながら長方形を描くと、長方形の範囲から飛び出すことなく収まっているオブジェクトだけが選択されます。

### オブジェクトのレイアウトを変更する

ラベルエディターでは、プレビューエリアに追加されたオブジェクトを変更することができます。すべてのオブジェクトに対し、サイズ変更、移動、整列、均等配置、レイヤ移動の操作が可能です。これらの変更はフィールドとグラフィックオブジェクトのどちらに対しても適用できます。

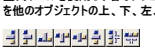
#### オブジェクトを移動する

1つ以上のオブジェクトを選択してマウスでドラッグすることにより、オブジェクトを移動することができます。また、矢印キーを使って移動することもできます:

- 一度に1ピクセルずつオブジェクトを移動するには、オブジェクトを選択して矢印キーを押します。
- 一度に10ピクセルずつオブジェクトを移動するには、オブジェクトを選択して、**Ctrl**キー (Windows) または**Command**キー (Macintosh) を押しながら矢印キーを押します。

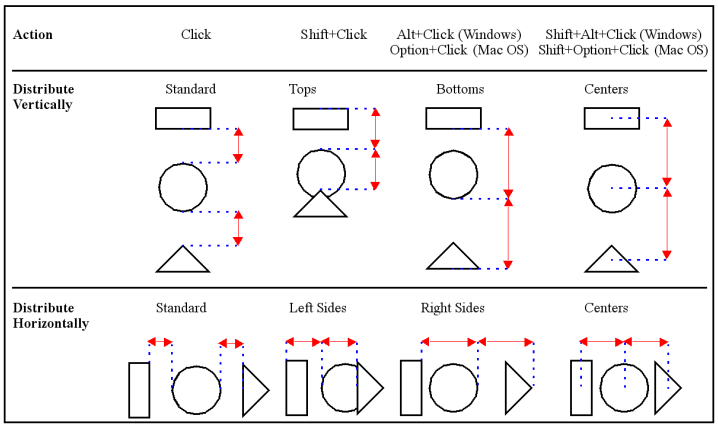
#### オブジェクトの整列と均等配置

整列ツールを使用し、各オブジェクトを互いに対して整列させることができます。あるオブジェクトをもう一つのオブジェクトと揃える場合、そのオブジェクトを他のオブジェクトの上、下、左、右、および縦中央、横中央の位置に揃えることができます。



"均等配置"とは、3つ以上のオブジェクトに対し、それらオブジェクトの間隔が縦、あるいは横方向に均等になるよう、自動調整する機能をいいます。多数のオブジェクトを扱う際に、特に便利となる機能です。自動調整の結果表示される間隔は、選択オブジェクト間の距離の平均です。

デフォルトでは、各オブジェクトの外周と外周を基準とした間隔を均等化します。下図のように、オブジェクトの上辺、下辺、中央を基準に均等配置することも可能です:




#### オブジェクトを重ねる

さまざまなレイヤ (層) でオブジェクトを使用するデザインを作成することができます。例えば、ラベル上でフィールドの背面に陰影付きの矩形を配置することができます。ラベルウィザードには**背面へ**と**前面へ**ツールがあり、このツールを使用してラベル上の各オブジェクトを重ねることができます。

**背面へ**または**前面へ**ツールをクリックすると、選択オブジェクトを最前面または最背面に移動することができます:

1レベルだけ前または後ろの層にオブジェクトを移動したい場合は、**Shift**キーを押しながら、**背面へ**または**前面へ**ツールをクリックします。

### オブジェクトの見た目を変更する

プレビューエリアに追加したオブジェクトの見た目を変更できます。例えば、オブジェクトの色やパターンを変えたり、テキストオブジェクトのフォントやサイズを変更することができます。

**注:** クリップボードからペーストした画像は変更できません。例えば、色の調整はラベルエディターでは行えません。変更できるのは、エディターの機能を使って描画したオブジェクトだけです。

#### オブジェクトサイズを変更する

ラベル上のオブジェクトを選択し、その選択ハンドルをドラッグすることにより、オブジェクトのサイズを変更することができます。また、矢印キーを使って操作することもできます:

- 一度に1ピクセルずつオブジェクトを拡大、または縮小するには、オブジェクトを選択し、**Alt**キー (Windows) または**Option**キー (Macintosh) を押しながら矢印キーを押します。
- 一度に10ピクセルずつオブジェクトを拡大、または縮小するには、オブジェクトを選択し、**Alt+Ctrl**キー (Windows) または**Option+Command**キー (Macintosh) を押しながら矢印キーを押します。

#### 描画色、背景色、塗りつぶしパターン、境界線の追加

**オブジェクトルック**エリアの機能使って、オブジェクトの背景色、ぬりつぶしパターン、境界線を設定できます:

- **背景色**、および**描画色**パレットからオブジェクトの背景、描画に使用する色を選択できます。オブジェクトがフィールドまたはスタティックテキストである場合、描画色はテキストの色を制御し、背景色はオブジェクトの矩形の色を制御します。エディターで描画された図形を選択した場合には、描画色は図形の境界線やパターンの色を制御し、背景色は図形の"白地"部分の色を制御します。
  **注:** 描画色や背景色をオブジェクトの境界線にのみ適用するには、**Shift**キーを押しながらオブジェクトを選択します。
- **塗りつぶし**パレットを用いて、選択オブジェクトの塗りつぶしパターンを制御します。また、**境界線**パレットにより、境界線の塗りつぶしパターンを制御します。これらはフィールドやスタティックテキストに対しては無効です。
- **線幅**ポップアップメニューは、二次元オブジェクトの線幅または境界線の幅を制御します。

ラベルにオブジェクトを追加すると、そのオブジェクトには直前に作成されたオブジェクトのオブジェクトルック属性が受け継がれます。そのオブジェクト (または任意のオブジェクト) にデフォルトのオブジェクトルック属性を割り当てたい場合は、オブジェクトを選択して**デフォルトルック**ボタンをクリックします。すると、描画色や背景色、塗りつぶしパターン、境界線の幅がデフォルト設定にリセットされます。

#### テキストの書式を変更する

フォントやフォントサイズ、行揃えや表示フォーマットなど、選択したテキストの見た目は**テキスト**エリア、**書体**エリアの設定を使って変更することができます。これらのパラメーターはフィールドとスタティックテキストに対してのみ有効です。

- **フォント**および**サイズ**メニューを使い、選択テキストに対してこれらの設定を行えます。メニューには、お使いのOSにて使用可能な選択肢が表示されます。フォントサイズは直接入力して指定することもできます。
- **行揃え**メニューを使い、タイトルの外枠に対する行揃え属性を変更することができます。
- **表示フォーマット**コンボボックスを使い、選択したフィールドに対して表示フォーマットを指定することができます。ラベルが印刷されるとレコードの値はこのパラメーターに基づいて表示されます。
  適用可能な表示形式は選択されたフィールドの型によります。また、コンボボックスに直接入力することで、独自のフォーマットを作成することもできます。表示フォーマットについての詳細は、表示フォーマットを参照ください。
- **書体**エリアでは、選択したテキストに対して標準の書体オプションを適用することができます。

## ラベルシートの作成

ラベルエディターを使って、ラベルを印刷するシートのレイアウトを指定することができます。ラベルシートのサイズ、ラベルのサイズ、シート当たりのラベル数、シートの向き、マージンといった項目の設定が可能です。

使用するラベルシートのレイアウトを指定するには、次の手順に従ってください。

1. **レイアウト**ページタブをクリックします。
2. アイコンをクリックして、ラベルシートの方向を指定します。
   ラベルシートの方向は用紙方向に左右されないという点に留意してください。用紙方向を変更するには、**プリント設定** ボタンをクリックし、プリント設定ダイアログボックスで用紙方向を変更します。変更された設定はラベルプレビューエリアの表示で確認できます。
3. 一般的に市販されている標準規格のラベルシートを印刷に使用する場合は、**ラベルテンプレート**ドロップダウンリストからラベル用紙のタイプを選択します。
   レイアウトページ上の他の入力エリアは、選択したラベル用紙の規格を反映する内容に変わります。詳細については、ラベルテンプレートを参照ください。必要に応じて、これらの設定を修正することができます。変更の必要がなければ、手順の9番に進んでください。
4. ラベルシートの各行のラベル数を**ラベル列数** ボックスに入力し、各列のラベル数を**ラベル行数** ボックスに入力します。
   ボックスに数値を直接入力することもできますし、ボックスに付いている矢印ボタンで数字を変更することも可能です。数値を確定するには、他のエリアをクリックするか、**Tab**キーまたは**Return**キーを押します。
   **自動サイズ調整**オプションを選択している場合に、ラベルの列数および行数を変更すると、各ラベルのサイズがシート上で最大となるよう、自動的に変更されます。ラベルプレビューエリアでは、ラベルの印刷ページ上の状態が更新されます。
5. 必要に応じて、マージンサイズの入力時に使用する計測単位を**単位** ドロップダウンリストから選択します。
   このパラメーターを変更すると、長さを表す数値はすべて、指定した単位に変換されます。
6. **上マージン**、**左マージン**、**水平方向間隔**、**垂直方向間隔** ボックスに値を入力して、ラベル用紙のマージンを指定します。
   マージンはラベルシートに対して適用され、指定した間隔はラベルとラベルの間に挿入されます。
7. **ページサイズ**ラジオボタンをクリックすると、下マージンと右マージンも指定できます。これらのマージンはプレビューエリアにおいて青い線で表示されます。
8. **ラベルサイズ**ラジオボタンをクリックすると、ラベルの幅と高さを指定できます。ラベルサイズは自動的に計算させることも可能です。この機能については、自動サイズ調整を参照ください。
9. 各ラベルを複数印刷したい場合は、**ラベル数／レコード**ドロップダウンリストを使用して、印刷する数を選択します。
   ラベルは用紙に連続して印刷されます。4Dはラベルページ全体を複製するわけではありません。
10. 印刷に使う1枚目のラベル用紙が部分的に使用されている場合には、ラベルプレビューエリアで1番目の空白ラベルをクリックします。4Dは指定されたラベルからラベルの印刷を開始します。この設定は、1枚目のラベル用紙のみ対象となります。
11. ラベルの印刷時にメソッドを実行したい場合は、**適用するメソッド**ドロップダウンリストからメソッドを選択します。
    この機能については、ラベルにメソッドを適用するを参照ください。

ラベルシートの設定が完了しました。ラベルの内容が、設定したラベルシートに収まるかどうかをチェックするには、**ラベル**タブをクリックします。

### ラベルテンプレート

**ラベルテンプレート**ドロップダウンリストから一般的に市販されているラベルテンプレートを選択することで、ラベルサイズやページ寸法、マージンなどを一括指定することができます。

ドロップダウンリストには4Dにインストールされている標準テンプレートが表示されます。テンプレートを選択すると、レイアウトページ上の他の入力エリアは、選択したラベル用紙の規格を反映する内容に変わり、印刷の準備が整います。レコード毎のラベル印刷枚数と、適用するメソッドに関しては、追加で指定することができます。

### 自動サイズ調整

ラベルの幅と高さは手動で指定することもできますが、他のパラメーターをもとにラベルエディターに計算させることもできます。この場合、4Dは1シート当たりの印刷ラベル数がなるべく多くなるように、スペースを活用します。

**自動サイズ調整**を選択すると、ラベル幅およびラベル高さは次のパラメーターに基づいて計算されます:

- ラベルの列数および行数
- 上マージンおよび左マージン
- 水平方向間隔および垂直方向間隔

**注:** 右マージンと下マージンが指定されていても、計算には考慮されません。

**自動サイズ調整**を選択している場合に、これらのパラメーターを変更すると、プレビューエリアに表示されたラベルのサイズが変更されます。

### ラベルにメソッドを適用する

ラベルには、印刷時に実行されるメソッドを適用することができます。ラベル毎またはレコード毎のいずれにメソッドを実行するか指定できます。

ラベルシートにメソッドを適用するには、**適用するメソッド**ドロップダウンリストから選択する必要があります。このドロップダウンリストにはデータベースにて作成されたすべてのプロジェクトメソッドが表示されます。メソッドを作成していなければ、リストは空です。

**注:** ドロップダウンリストにはコンポーネントメソッドや、非表示設定されたプロジェクトメソッドは表示されません。

メソッドを選択したら、**実行タイミング**ラジオボタンを使って、**ラベル毎**または**レコード毎**のいずれかの実行タイミングを指定します。レコード毎のラベル印刷枚数が1の場合には、どちらを選んでも違いはありません。

# データの読み込みと書き出し


概要

ファイルフォーマット

フィールド、レコード、ファイルの区切り文字

ファイルからデータを読み込む

ファイルへデータを書き出す

ODBC データソースを使用した読み込みと書き出し

読み込み設定と書き出し設定の保存とロード

## 概要

4Dのデータの読み込みと書き出し機能は、データベースからデータの受け渡しを実行できる、高速で信頼性の高い方法を提供します。他のデータベースやスプレッドシートから4Dへアップグレードする場合は、データを読み込むことにより、キーボードからのデータ再入力作業が不要になります。
Windows (Macintosh) データの管理用アプリケーションや、Windows (Macintosh) 以外のアプリケーションからWindows (Macintosh) へ転送されたデータを読み込むことができます。

また、4Dからデータを書き出し、データ処理を行う他のプログラムでそのデータを使用することも可能です。例えば、グラフィックや統計専門のプログラムへデータを書き出すことができます。さらに、書き出したデータを別のコンピューターへ転送することもできます。

データの読み込みや書き出しは、該当するダイアログボックスの設定を使用したり、またはフォームを使用して行います。ダイアログボックスによる読み込みや書き出しは、フォームを使用した場合よりも高速に実行されます。しかし、フォームを使用すると、オブジェクトメソッドやフォームメソッドを用いてデータを処理することができます。つまり、フォームを使用してデータの読み込みや書き出しを行うと、処理される各レコードごとにフォームメソッドとオブジェクトメソッドが実行されます。

読み込みや書き出しの設定を指定した後は、その設定を読み込み／書き出し設定用ファイルへ保存したり、またはロードすることができます。

最後に4Dでは、中間ファイルを用いずに、ODBC データソースからのデータ読み込みや、データソースへのデータ書き出しを行うことができます。この件については、ODBC データソースを使用した読み込みと書き出しで説明しています。

**注：**

- クィックレポートエディターを使用して、レコードを書き出すこともできます。詳細については、クイックレポートの生成のディスクファイルを参照してください。
- 4Dのランゲージコマンドを使用して、レコードを書き出しすことや読み込むことが可能です。詳細については4D *ランゲージリファレンス*マニュアルのインポート＆エクスポートを参照してください。

# ファイルフォーマット

ファイルフォーマットは、読み込みファイルや書き出しファイル内での、フィールドやレコードの格納方法を示します。ファイルフォーマットの中には、フォーマット情報を含むものもあります。4Dでは、いくつかのファイルフォーマットがサポートされており、更にいくつかの内部フォーマット機能を提供しています。

## サポートされているフォーマット

4Dでは次のファイルフォーマットがサポートされます。

### テキスト (ASCII)

このフォーマットは、レコード内の各フィールドを"フィールド終了"区切り文字で区切り、各レコードを"レコード終了"区切り文字で区切ります。フィールドやレコードの区切り文字のデフォルト値は、それぞれタブとリターンです。
このフォーマットは特定の文字セットを持たないため、**文字セット**メニューにて任意の文字セットを選択できます (デフォルトは UTF-8です)。また、USE CHARACTER SETコマンドを使って指定することもできます。UTF-16はサポートされていません。

文字セットをUTF-8に設定すると、4Dはテキストへの書き出しの際に Byte Order Mask (BOM) を追加します。読み込みの際に BOM を探知した場合には、BOM によって指定された文字セットが優先されます。

### 固定長テキスト

固定長テキストフォーマットは、一定の文字数を各フィールドに割り当てます。フィールド終了区切り文字は使用されません。それよりも少ない文字数でレコードの値が収まる場合でも、すべてのレコードは同じ文字数になります。割り当てられた文字数よりもレコードの値が短い場合は、充てん文字 (通常はスペース) を用いてフィールドが埋められます。各データタイプに対して充てん文字を指定するか、またはデフォルトの文字を使用することができます (埋め文字文字ページを参照してください)。一方で、フィールドに割り当てられた文字数よりも値の文字数が多い場合は、その値が切り捨てられます。

固定長テキストを読み込む際に、4Dは充てん文字を消去します。

このフォーマットのその他の特性については、テキストフォーマットと同様です。

### DIF

Data Interchange Formatフォーマット (データ交換フォーマット) の略語です。

DIFはデータ交換のための標準であり、規則として"IBM437"セットを使用します。この文字セットは別名"Latin-US (DOS)"とも呼ばれます。この名前はCONVERT FROM TEXTとConvert to textコマンドでサポートされます。このフォーマットを選択すると、4DはUSE CHARACTER SETコマンドで設定された読み込み/書き出しフィルター設定を無視します。

プラットフォームを選択するとレコード区切り文字のみを決定します。WindowsではCR/LF、Mac OSではCRです。

DIFフォーマットでは文字データ中に改行 (文字コード0x0Aと0x0D) を含むことができません。書き出し時に4Dはこれらの文字をスペースに置き換えます。さらにダブルクォーテーションが内部的な区切り文字として使用されます。文字データ中のクォーテーションマークは、ふたつの " にエスケープされて書き出されます。
例えば Monday\rTuesday"Wednesday" は "Monday Tuesday""Wednesday"" となります。

データを読み込む際、4Dは文字列の最後にある埋め文字を取り除きます。

### DBF

dBase は、DOS およびWindows のデータベースアプリケーションで一般的に使用されるファイルフォーマットです。

4DはDIFフォーマットと同様にこのフォーマットを処理します。

DBFフォーマットの場合、文字列は埋め文字を使用して固定長で書き出されます (データ型に応じて' 'または'0')。データ読み込みの際、データの後ろにある埋め文字は取り除かれます。

### SYLK

SYmbolic LinK フォーマットの略語です。

このフォーマットには特定の文字セットは指定されていません。必要に応じて文字セットメニュー (デフォルトはUTF-8) またはUSE CHARACTER SETコマンドを使用して選択できます。Byte Order Markは使用されません。UTF-16はサポートされません。

レコード区切り文字はプラットフォームに応じます。WindowsではCR/LF、Mac OSではCRです。

書き出される文字列中に含まれる改行 (0x0D) は以下のようにエスケープされます: Windowsでは *<escape><blank>*:、Mac OS *では<escape><blank>*= 。

**注:** 4D Viewプラグインも読み込みと書き出しでSYLKフォーマットをサポートします。

### 4Dフォーマット

4D固有のフォーマットであり、このフォーマットにより、異なる4Dデータベース間でのレコード交換を容易に行うことができます。このフォーマットでは、ピクチャーやBLOB をはじめ、すべての4thDimension データタイプがサポートされます。4D書き出しファイルの拡張子は".4IE"です。

読み込みや書き出しを行う際、ファイルがMacintosh またはWindows 環境のいずれに対応するかの指定はできなくなりました。また、文字セットを指定することもできません (Unicodeが適用されます)。読み込みおよび書き出しフィルタでUSE CHARACTER SETコマンドを実装しているものは無視されます。

### XML

XML (eXtensible Markup Language) はデータ変換の規格です。この言語はタグの使用を基本とし、交換されるデータとその構造は、このタグを用いて正確に記述することができます。XML ファイルはテキストフォーマットのファイルであり、その内容はデータを読み込むアプリケーションにより解析されます。大多数のアプリケーションがこのフォーマットをサポートしています。XML の用語に関する詳細は、4D *ランゲージリファレンス*マニュアルのXMLを参照してください。

このフォーマットは特定の文字セットと関連しておらず、任意の文字セットを書き出しダイアログボックスにて指定できます。XMLファイル内に文字セットの指定が含まれます。USE CHARACTER SETコマンドを用いて指定した読み込み/書き出し用文字セットは無視されます。

"レコード終了"区切り文字はプラットフォームによって違います: Windows ではReturn+ラインフィード、Macintosh ではReturnです。

## フォーマットオプション

読み込み、あるいは書き出しを行うファイルフォーマットについて、次のオプションが選択できます。

### 文字セットの選択

テキスト、SYLK、そしてXML (書き出し) のファイルフォーマットを扱う場合には、**文字セット**メニューを利用できます。このメニューには IANA で定義された標準的な文字セットのリストが含まれます (詳細な情報は次ののWeb サイトを参照してください:
http://www.iana.org/assignments/character-sets)。

**注:** DIF およびDBF フォーマットでは**文字セット**メニューが"IBM437" に固定されます。またこのメニューは4Dフォーマットでは使用できません。

- 書き出しの場合、このメニューを使用してデータ書き出しに使用する文字セットを指定できます。
- 読み込みの場合、このメニューを使用して、読み込まれたデータのエンコーディングを指定できます。書き出しファイルのヘッダーににBOM が含まれる場合 (ヘッダーページ参照)、読み込みデータのエンコーディングは定義済みとなるため、このメニューは利用不可になります。
  読み込みおよび書き出し処理における、デフォルトのエンコーディングはUTF-8か、あるいはUSE CHARACTER SETコマンドが実行されている場合には、それによって指定された文字セットになります。読み込み/書き出しダイアログボックスにて特定のエンコーディングを指定しても、アプリケーション自身のカレント文字セットに影響はないことに注意してください。

### 書き出し先プラットフォーム

このメニューを使って、フィールド、レコード、ファイルの区切り文字を設定することができます。次のオプションから選択できます:

- 自動: 書き出しプラットフォームに基づいて、レコード終端文字を設定します。
- Macintosh: フィールドの終わり = Tab、レコードの終わり = 改行、ファイルの終わり = なし。
- Windows: フィールドの終わり = Tab、レコードの終わり = 改行+ラインフィード、ファイルの終わり = なし。
- Unix: 終端文字 = ラインフィード。
- カスタム: 区切り文字ページを表示します。

## フィールド、レコード、ファイルの区切り文字

フィールド終了区切り文字はレコードのフィールド間に置かれ、レコード終了区切り文字は各レコードの後ろに置かれます。ファイル終了区切り文字は、読み込み／書き出しファイルの最後に置かれます。特定のアプリケーションとファイルを交換する際に、これらの区切り文字が必要になる場合があります。

区切り文字は、テキスト (ASCII) と固定長テキストフォーマットでのみ使用されます。しかし、固定長テキストフォーマットではフィールド区切り文字が使用されず、一般的にテキストフォーマットでも明示的なファイル区切り文字は使用されません。

テキストフォーマットでデータを読み込む際、4Dはファイルに埋め込まれた区切り文字を使用して、フィールドとレコードの終わりを判断します。また、データを書き出す際には、4Dはこれらの区切り文字をフィールドへ自動的に挿入します。

フィールドやレコードの区切り文字としても使用される文字が、フィールドに含まれていないことを確認してください。例えば、レコード区切り文字として改行が使用されている場合、フィールドに改行が含まれていてはいけません。データを読み込む際に、4Dはすべての改行をレコード区切り文字であるものと解釈します。したがって、フィールド内の改行は、レコードの終わりとして間違って解釈されてしまいます。不適切な場所に改行を入れると、読み込み処理が正しく行われません。テキストフィールドを含むデータを書き出す場合、テキストフィールド内に改行が含まれているかもしれません。この可能性がある場合には、データを書き出す前に、別のレコード区切り文字を選択するか、または含まれている改行を削除してください。

## ファイルからデータを読み込む

ファイルのデータは、XML、4D、SYLK、DIF、DBF、テキスト、または固定長テキストフォーマットで読み込むことができます (ファイルフォーマット参照)。他のアプリケーションから書き出されたデータを読み込む場合は、そのアプリケーションのドキュメントを参照して、これらのフォーマットのいずれかでデータを書き出す際の情報を調べてください。他のアプリケーションがデータ書き出しの際に異なるフォーマットを利用している場合には、テキストエディターやワードプロセッサーを使用して、先にファイルを修正しておく必要があるかもしれません。

使用するデータベースと同じプラットフォーム上で動作している別のアプリケーションからデータを読み込むには、まず最初に別のアプリケーションを使用してデータを書き出します。
その際に注意すべき項目は、フィールドの書き出し順序と、そのアプリケーションがデータ保存に使用するフォーマット、そして、テキストフォーマットを用いる場合には、そのアプリケーションで使用される区切り文字です。データを読み込む際に、これらの設定を一致させなければなりません。

データの読み込みを行う前に、すべてのテーブルストラクチャーを構築しておく必要はありません。読み込みダイアログボックスから、必要なテーブルを直接作成することができます。

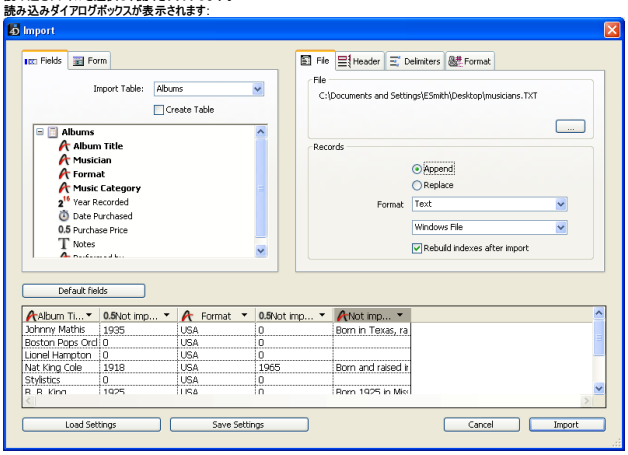
読み込みダイアログボックスでは、読み込まれるデータに対応するフィールドを選択したり、または読み込まれるデータ用のフィールドを含む入力フォームを指定することができます。

データの読み込みは、デザインモードにおいては読み込みダイアログボックスを介して行います。アプリケーションモードにおいては、IMPORT DATAコマンドを使ってこのダイアログボックスを表示させることも出来ますが、インポート＆エクスポートテーマのコマンドを使って直接ファイルを読み込むこともできます。

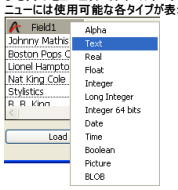
### 読み込むデータを指定する

デザインモードでデータを読み込むには、次の手順に従ってください:

1. 4Dの**ファイル**メニューから**読み込み＞ファイル選択...**を選択します。
   4Dはファイルオープン用のダイアログボックスを表示します。読み込まれるファイルのタイプ (XML、TEXT、DIF、DBF、SYLK、4IE) を選択することができます。ファイルタイプに関する詳細は、ファイルフォーマットを参照してください。
2. **読み込むファイルを選択し、開く**をクリックします。
   **読み込みダイアログボックスが表示されます**:




   ファイルエリアには、読み込まれるファイルのパス名が表示されます。別のファイルを読み込みたい場合は、**参照**ボタンをクリックしてファイルを選択してください。
3. テーブルとフィールドの選択エリアにおいて、データを読み込みたいテーブルとフィールドを選択します。
   次のオプションから1つを選択することができます:
   - **既存のテーブルとフィールドへデータを読み込む**: 読み込みテーブルドロップダウンリストからテーブルを選択します (デフォルトでは最初のテーブルが選択されています)。階層リストを使用して、データを読み込むフィールドの表示や選択を行えます。
   - **読み込むデータ用に新規テーブルを作成する**: **テーブル作成**チェックボックスを選択すると、4Dは新しいテーブルを作成し、そこへデータを読み込みます。4Dは、読み込まれるデータに従ってフィールドの数やフィールドのタイプを決定します。新しいテーブルやフィールドの名前を変更したい場合は、テーブルとフィールドの選択エリアでそれをダブルクリックし、編集可能な状態にします。
     テキストファイルに列タイトルがデータとして含まれている場合、**フィールド名を列タイトルにする**チェックボックスを選択することで、それらの名前をフィールド名として使用できます。(ヘッダページ参照)
     また、プレビューエリアでフィールドのタイトルバーをクリックすると、新規フィールドのデータタイプを変更することもできます。ポップアップメニューには使用可能な各タイプが表示されます:




     テーブルは、実際の読み込み処理の際に作成されます。読み込みをキャンセルするか、またはこのオプションの選択を解除すると、データベースストラクチャーは変更されません。
     **注:** パスワードアクセスシステムを使用する場合、**テーブル作成**オプションオプションを無効にして、ユーザに読み込みダイアログボックスから新しいテーブルをデータベースに作成させないようにすることができます。このオプションを無効にするには、環境設定ダイアログボックスを使用して、デザインモードへのアクセスが可能なアクセスグループを作成します (もちろん、まず4Dのパスワードアクセスシステムを有効にしておく必要があります)。このアクセスグループに属さないユーザは、読み込みダイアログボックスの**テーブル作成**オプションを選択することができません。
   - **フォームを使用してデータを読み込む**: このオプションに関しては、フォームを使用してデータを読み込むで説明しています。
   プレビューエリアには、読み込みファイルの内容が表データとして表示されます。フォームを使用してデータを読み込む場合、フォームの全フィールドが表示されます。各カラムのサイズは、マウスでタイトルバーの境界線を選択してドラッグすると変更することができます。
4. 必要に応じて、プレビューエリアでヘッダーエリアのポップアップメニューを使用し、読み込みファイルの各カラムが読み込まれるフィールドを選択します。
   **注:** テーブル作成オプションを選択している場合や、フォームを用いてデータを読み込む場合は、この機能を使用できません。
   各カラムのタイトルバーには、そのカラムが読み込まれるフィールド名とそのデータタイプが示されます。必要があれば、ポップアップメニューを使用して、別のフィールドを選択することができます。

   フィールドにカラムを割り当てるには、まず任意のカラムを選択し、次に階層リストのフィールド名をダブルクリックする操作方法もあります。
   また、**読み込まない**オプションを選択することも可能です。その場合、データのカラムは読み込まれません。
   - **フィールド自動割当**
     このボタンを使用すると、カラムの表示順ではなく、そのタイプに従って、読み込みファイルのカラムをテーブルのフィールドへインテリジェントに割り当てることができます。各カラムのデータタイプが推定され、そのタイプに対応するフィールドへカラムが割り当てられます。この推定は次の方法で行なわれます:

| 推定されるフィールドタイプ | 4Dのタイプ |
|---|---|
| すべての数値 | 実数 (数値) |
| True／False | ブール |
| 4Dフォーマットの1つでの日付 | 日付 |
| 時間 | 時間 |
| その他 | 文字 |

     対応するフィールドが見つからなければ、そのカラムは読み込まれません。このカラムを読み込むには、プレビューエリアにてヘッダーエリアのポップアップメニューを使用し、手動でカラムをフィールドに割り当てなければなりません。
   - **固定長テキストファイルフォーマットの文字数**
     固定長テキストのファイルを読み込む場合、プレビューエリアでは、そのフィールドのデータタイプアイコンの下に各カラムの文字数が表示されます。最初の行のデータに基づいて、各カラムの文字数が割り当てられます。ヘッダエリアのカラムディバイダーをドラッグするか、またはフィールド長エリアに値を入力して、各カラム間の文字の配分を変更することができます。ただし、文字を行に追加することはできません。
5. (任意) レコードエリアのオプションを選択します。オプションの詳細については、レコードエリアで説明しています。
6. (任意) ヘッダー、区切り文字、充てん文字、XML、フォーマットの各ページを使用して、その他の読み込みオプションを選択します。
   これらのオプションについては、読み込みオプションで説明しています。
7. 設定を保存したい場合は、設定を保存ボタンをクリックします。
   このオプションについては、読み込み設定と書き出し設定の保存とロードで説明しています。

8. **読み込み** ボタンをクリックして、テキストファイルの読み込みを開始します。
   4Dはデータベースへデータを読み込んでいる間、進捗インジケータを表示します。

**読み込みオプション**

4Dでは、さまざまな読み込みオプションが提供され、オプションエリアのタブを使用して、これらのオプションにアクセスすることができます。タブの数 (1つから5つ)、およびその内容は、使用するファイルのタイプや選択したオプションに応じて変わります。

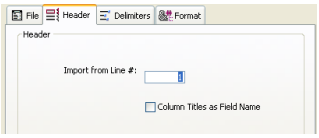
**レコードエリア**

**ファイル**ページのレコードエリアにて選択できるオプションでは、読み込み処理の基本設定を行えます:

- **追加**、または**置換**: 読み込まれたレコードでカレントセレクションを置き換えるか (**置換**オプション)、または既存のデータに追加するか (**追加**オプション) を指定します。
- **フォーマット、文字セット、書き出し先プラットフォーム**: これらは読み込みファイルのフォーマットに関するオプションです。(詳しくは、ファイルフォーマット参照)
- **読み込み後のインデックスの再構築**: このオプションにチェックがついている時 (デフォルト)、インデックスはデータの読み込みが完了した後に再構築されます。このメカニズムは大量のデータの読み込み速度を改善するのに有効です。フィールドの既存データに比べ、読み込むデータ量が少量である場合には、読み込み前にこのオプションのチェックを外すとよいかもしれません。この場合、インデックスは徐々に更新され、完全な再構築はされません。

**ヘッダページ**

このページは、テキストファイルまたは固定長テキストファイルから読み込む場合にのみ使用できます。



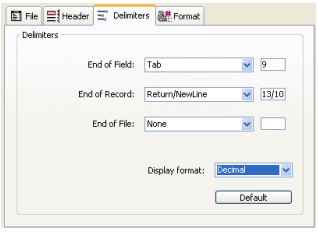

- **次の行番号から読み込み**: このオプションを使用して、読み込まれるテキストファイルの最初の行を指定することができます。読み込みファイルのヘッダ情報 (タイトルやフィールド名など) を読み飛ばすよう4Dに指示するには、この機能を使用します。読み込む各カラムは、最初の行のフォーマットに従って推定されるため、読み込みファイルがフォーマットされていない行 (タイトル、日付など) から開始する場合には、このオプションが非常に役立ちます。

- **フィールド名を列タイトルにする**: このオプションは、**テーブル作成**オプションを選択している場合に、カラムタイトルをフィールド名として使用するよう4Dに指示します。

必要に応じて、これらの2つのオプションを一緒に使用してください。例えば、読み込みファイルがタイトルで始まり、その次にフィールド名、また次に最初のデータ行と続く場合、次の行番号から読み込みのパラメータを2に設定し、次にフィールド名を列タイトルにするチェックボックスを選択します。

**区切り文字ページ**

このページを使用して、テキストファイルと固定長テキストファイルで使用されるフィールド、レコード、ファイルの区切り文字を指定することができます。このオプションはレコードエリアの**書き出し先プラットフォーム**メニューのレコードエリア設定によって予め自動設定されています。

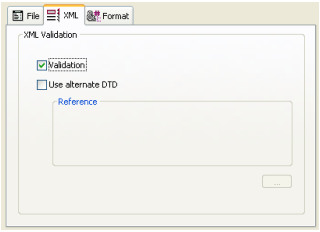



区切り文字 (デリミター) については、フィールド、レコード、ファイルの区切り文字で詳しく説明しています。デフォルトの区切り文字を変更するには、ドロップダウンリストに表示される、各区切り文字として最も頻繁に用いられる値から選択します。また、右側の入力エリアに区切り文字を直接入力して指定することも可能です。
ページ下部にあるドロップダウンリストにより、別の形式 (**文字**、**10進数** (デフォルト形式)、**16進数**) で区切り文字を表示することができます。ただし、文字形式を選択した場合に、区切り文字が印刷不可文字 (タブ、ラインフィード、リターンなど) だと、何も表示されない点に注意してください。

**注:** 区切り文字は、**デフォルト設定**ボタンをクリックするか、**ファイル**ページでドキュメントのタイプを変更すると、デフォルト値にリセットされます。

**XML ページ**

このページは、XML ファイルの場合にのみ使用することができます。このページを使用して、読み込まれるXMLコンテンツの解析モードを設定します。




データの読み込み時に、4DはXML 文書の情報を取り出すために、その内容を処理します。デフォルトでは、特に妥当性を検証せずにこの処理が実行されます。XML文書は"整形式"、つまりその構造は正しく、解析における曖昧さがないものとみなされます。

しかし、読み込み時に文書の"妥当性検証"を要求することができます。これを行うには、**検証**オプションを選択してください。この場合、4Dは文書のDTD (Document Type Definition : 文書型定義) に基づいて文書内容を解析し、その内容が定義と一致しているかどうかを調べます。文書が妥当である場合にのみ、読み込みが行われます。

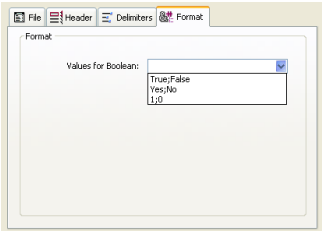
**注:** DTD に関する詳細は、XML ページを参照してください。

読み込まれる文書の DTD がその文書の中ではなく別のファイル内に納められている場合や、あるいは他の DTD を使用して文書の検証を行いたい場合は、**他の DTD を使用**オプションを選択し、**参照**ボタンを用いて DTD を含むファイルを指定します。

読み込みが終了するとダイアログボックスが閉じられ、データが読み込まれたテーブルがカレントテーブルになります。

**フォーマットページ**

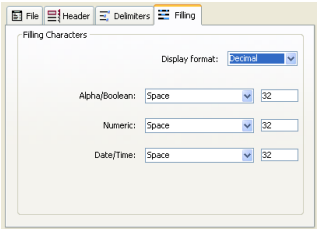
このページは、テキストファイル、固定長テキストファイル、XML ファイルで使用できます。このページを使用し、読み込まれたブールフィールドのフォーマットを設定することができます。




使用する構文は、"*True (真) の値; False (偽) の値*"です。例えば、ブールフィールドをカラムに読み込み、その値が"黒" (True の値) と"白" (False の値) である場合は、"黒;白"と入力します。コンボボックスには、最も一般的に使用されるブール用フォーマットが表示されます。コンボボックスに存在しない任意のフォーマットも入力可能です。

**埋め文字文字ページ**

このページは、固定長テキストファイルにのみ表示されます。このページを使用し、それぞれのデータタイプに対して使用する埋め文字を定義することができます。デフォルトの埋め文字はスペース (空白) です。



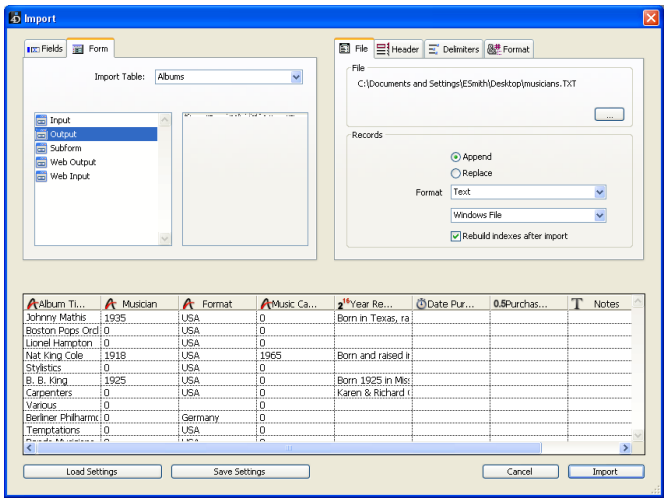

固定長テキストファイルフォーマットに関する詳細は、ファイルフォーマットを参照してください。

**文字／ブール** (ブールフィールドは文字として読み込まれます)、**数字** (整数、倍長整数、実数)、**日付／時間**という3つのデータタイプに対して、異なる埋め文字を使用することができます。通常、固定長形式のフィールドを埋める場合は空白文字が用いられます。ドロップダウンリストには、それ以外の選択肢が提示されています。右側の入力エリアには、選択した埋め文字が表示されます。他の文字を使用したい場合、その文字を入力エリアに直接入力してください。

このページの上部にあるメニューを使用すると、別のフォーマットで埋め文字を表示することができます。**文字**、**10進数** (デフォルトフォーマット)、**16進数**が使用可能です。文字フォーマットを選択した場合、印刷不可の埋め文字は表示されません。

**フォームを使用してデータを読み込む**

テキストファイルをフォームへ読み込みたい場合は、テーブルとフィールドの選択エリアの上部にあるフォームタブをクリックします。フォームを使用する利点は、各レコードが読み込まれる際に、フォームに関連付けられたフォームメソッドやオブジェクトメソッドが実行されることです。これにより、データを読み込みながら、データを処理することができます (つまり、独自の読み込みルーチンの作成は不要になります)。




入力可能な変数やフィールドが含まれるフォームを使用できますが、ボタンが含まれるフォームは使用できません。各レコードが読み込まれるたびに、フォームメソッドおよび変数やフィールドに関連付けられたメソッドが実行されます。しかし、これらのメソッドの結果はプレビューエリアに表示されない点に注意してください。

読み込み用のフォームを使用する場合、次の事項に留意してください:

- フォームの入力順により、データのカラムが読み込まれる順序が決まります。フォームの入力順とテキストファイルのカラムの順序が一致することを必ず確認してください。
- テキストファイルのカラム数よりフォーム上のフィールドや変数の数が少ない場合、余分なカラムは無視されます。
- 読み込み用に使用されるフォームにはボタンを置かないでください。また、サブフォームオブジェクトは無視されます。

**4D Server:** サーバー上のストアードプロシージャーのメソッドを含むフォームを使用して、データを読み込むことはできません。フォームイベントはサーバー上で管理されないため、割り当てたメソッドが呼び出されません。

## ファイルへデータを書き出す

データを書き出す際に、4D、XML、SYLK、DIFF、DBF、テキスト、または固定長テキストフォーマットのファイルを作成することができます。これらのファイルは、別のアプリケーションからオープンしたり、読み込むことが可能です。他のアプリケーションで使用するためにデータを書き出す場合は、そのアプリケーションのドキュメントを参照し、データ読み込みに関する情報を調べて、そのアプリケーションと互換性があるファイルフォーマットと区切り文字を選択してください。

場合によっては、目的のアプリケーション用に書き出しファイルを正確に作成するため、1つ以上の書き出しオプションを使用する必要があります。例えば、プログラムのなかには、テキストファイルフォーマットを受け入れるが、最初のレコードがフィールド名でなければならないものもあります。

この場合、オプションエリアのヘッダーオプションを使用して、このレコードを追加することができます。

4Dは現在の並び替え順でレコードを書き出します。テーブルの全レコードを書き出すか、またはカレントセレクションだけにするか選択することができます。

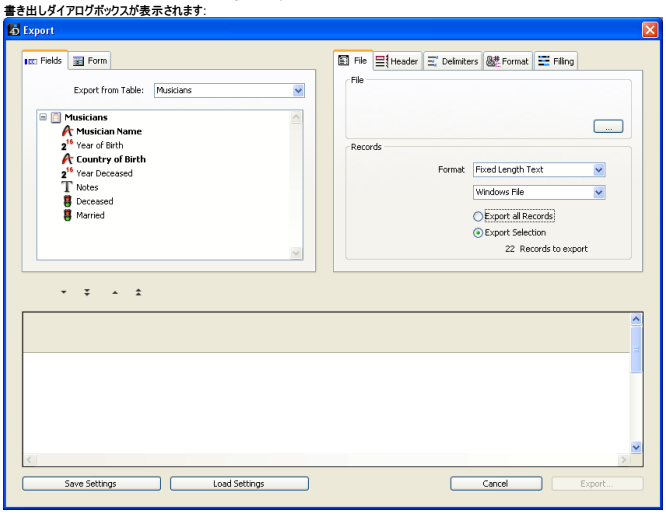
さらに、書き出すフィールドを指定するには、書き出しダイアログボックスで選択したり、またはフォームを選択することができます。フォームを使用すると、フォーム上のフィールドだけが書き出されます。

データの書き出しは、デザインモードにおいては書き出しダイアログボックスを介して行います。アプリケーションモードにおいては、EXPORT DATAコマンドを使ってこのダイアログボックスを表示させることも出来ますが、インポート＆エクスポートテーマのコマンドを使って直接ファイルを書き出すこともできます。

### 書き出すデータを指定する

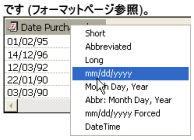
デザインモードでデータを書き出すには、次の手順に従ってください:

1. 4Dの**ファイル**メニューから**書き出し＞ファイル選択...**を選択します。
   書き出しダイアログボックスが表示されます:



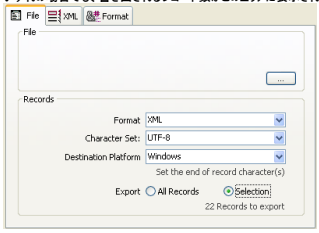

2. ファイルページの"レコード"エリアにて、書き出しファイルのフォーマットを選択します。**テキスト**、**固定長テキスト**、**DIF**、**DBF**、**SYLK**、**XML**、**4IE** のなかから選択し、必要に応じてそれらのフォーマットオプションも設定します (ファイルフォーマット参照)。
3. (任意) ファイルエリアの**参照...**ボタンをクリックし、書き出しファイルの名前と保存場所を入力して**保存**をクリックします。
   書き出しダイアログボックスが再び表示され、ファイルエリアに書き出しファイルのパス名が示されます。このステップでは、書き出し処理はまだ開始されません。(これらのステップを省略した場合は、**書き出し...**ボタンのクリック時にも、書き出しファイルの名前と保存場所を指定することができます)
4. テーブルとフィールドの選択エリアにおいて、書き出したいテーブルとフィールドを選択します。
   次のオプションのいずれかを選択することができます:
   - **書き出しダイアログボックスにおいて、書き出すフィールドを選択する。**
     このオプションを使用する場合、テーブルから書き出しドロップダウンリストからテーブルを選択して、書き出し対象となるフィールドをフィールドリスト上に表示させます。次に、プレビューエリアの真上にあるボタンを用いてフィールドを選択します:

   - **フォームを使用してデータを書き出す。**
     このオプションにより、フォーム上のフィールドだけが書き出されます。このオプションについては、フォームを使用してデータを書き出すで詳しく説明しています。
5. (任意) プレビューエリアにおいて、書き出されるデータのフィールドを変更します。
   **注:** フォームを使用してデータを書き出す場合は、この機能を使用できません。
   プレビューエリアには書き出すファイルの内容がカラムデータとして表示されます。各カラムのヘッダーエリアにはポップアップメニューがあり、それを使ってカラムに割り当てられたフィールドを変更することができます。カラムのサイズ変更も可能です。
   フィールド名の左にあるアイコンはデータタイプを示します。カラムヘッダー上で右クリックすると、そのカラムのデータタイプに対応する各フォーマットのコンテキストメニューを表示することができます。必要な場合は、このコンテキストメニューからフォーマットを選択してください。
   フォーマットを指定しない場合は、**デフォルトフォーマット**が使用されます。また、**フォーマット**オプションページでフォーマットを選択することも可能です (フォーマットページ参照)。




   書き出しファイルやプレビューエリアからカラムを削除することができます。削除を行うには、カラムのヘッダーをクリックしてから、**削除**ボタンをクリックします。
   - **固定長テキストのフィールドの長さを変更する**
     固定長テキストフォーマットを選択した場合は、プレビューエリアに各カラムの文字数が示されます。

     各カラムのサイズは、マウスでタイトルバーのカラムディバイダをドラッグして変更することができます。固定長テキストフォーマットを使用する場合は、この操作により、書き出しファイルのカラムに割り当てられる文字数が変わります。フィールドのデータタイプアイコンの下の入力エリアに値を直接入力して、文字数を指定することもできます。デフォルトの文字数は、テキストフィールドが80桁、文字フィールドはデザインモードで指定された最大長、数字フィールドは10桁です。
6. レコードエリアにおいて、選択したテーブルの全レコードを書き出すか (**全レコードを書き出し**オプション)、またはカレントセレクションだけを書き出すか (**セレクションを書き出し**オプション) を指定します。
   いずれの場合でも、書き出されるレコード数がこのエリアに表示されます。



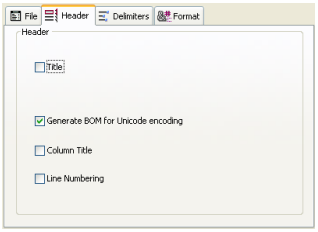

7. (任意) **ヘッダー**、**区切り文字**、**フォーマット**、**XML**、**充てん文字**ページを使用して他の書き出しオプションを指定します。
   これらのオプションについては、書き出しオプションで説明しています。
8. 設定を保存したい場合は、**設定保存**ボタンをクリックします。
   この機能に関しては、読み込み設定と書き出し設定の保存とロードで説明しています。
9. **書き出し**をクリックして、ディスクファイルへの書き出しを開始します。
   書き出しファイルのパス名を指定していない場合 (前述のステップ3) は、標準のファイル保存用ダイアログボックスが表示されます。設定済みの場合には、直接書き出し処理に移ります。

**書き出しオプション**

4Dでは、さまざまな書き出しオプションが提供され、オプションエリアのタブを使用して、これらのオプションにアクセスすることができます。タブの数、およびその内容は、使用するファイルのタイプや選択したオプションに応じて変わります。

**ヘッダーページ**

このページは、テキストファイルや固定長テキストファイルの書き出しにのみ使用できます。

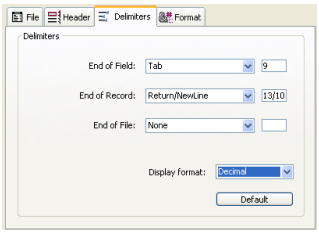



このページを使用すると、タイトル、フィールド名の行、レコード数を書き出しファイルに追加することができます:

- **タイトル**: このオプションにより、書き出しファイルにタイトルを出力することができます。チェックボックスの下のエリアにタイトルを入力します。
- **BOMを生成**オプション: このオプションは、書き出しファイルのヘッダーにByte Order Mark (BOM) を付加します。この追加の情報により、BOMをサポートするアプリケーションではテキストの解釈が容易になります。
  このオプションはデフォルトで選択されていますが、書き出しにUnicode 文字セットが選択されているときにのみ有効です。そうでない場合BOMは追加されません。
- **列タイトル**: このオプションは、書き出しファイルの最初の"レコード"としてフィールド名を書き出します。
- **行番号**: このオプションは、書き出されたレコードである、それぞれの行に番号を付けます。この番号は1から始まり1つずつ増加します。

**区切り文字ページ**

このページを使用して、テキストファイルと固定長テキストファイルで使用されるフィールド、レコード、ファイルの区切り文字を指定することができます。このオプションはレコードエリアの**書き出し先プラットフォーム**メニューのレコードエリア設定によって予め自動設定されています。

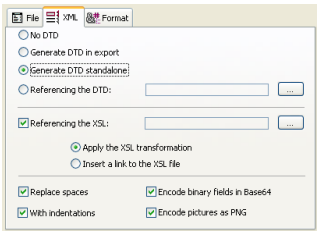



区切り文字 (デリミター) については、フィールド、レコード、ファイルの区切り文字で詳しく説明しています。デフォルトの区切り文字を変更するには、ドロップダウンリストに表示される、各区切り文字として最も頻繁に用いられる値から選択します。また、右側の入力エリアに区切り文字を直接入力して指定することも可能です。
ページ下部にあるドロップダウンリストにより、別の形式 (**文字**、**10進数** (デフォルト形式)、**16進数**) で区切り文字を表示することができます。ただし、文字形式を選択した場合に、区切り文字が印刷不可文字 (タブ、ラインフィード、リターンなど) だと、何も表示されない点に注意してください。

**注:** 区切り文字は、**デフォルト設定**ボタンをクリックするか、**ファイル**ページでドキュメントのタイプを変更すると、デフォルト値にリセットされます。

**XML ページ**

このページのパラメーターは、XML ファイルの書き出しにのみ使用できます。このページを使用し、書き出されるXML ファイルの内容を設定することができます。




**DTD**

XML フォーマットで書き出しを行う際、4Dでは DTD (*Document Type Declaration*) を生成するかどうかを選択することができます。DTD には、XML が従わなくてはならない一連の特定規則とプロパティが記録されます。より具体的に言えば、これらの規則により、それぞれのタグの名前と内容、およびその順序が定義されます。要素を形式的に定義することにより、XML 文書が"妥当"であることを確認でき、特に XML 文書内に繰り返し使用されるタグがある場合は、この規則が役立ちます。ただし、DTD は強制ではありません。
DTD の取り扱い方法を定義するために、次のオプションのいずれかを選択しなければなりません:

- **DTD なし** (デフォルトオプション): このオプションを選択すると、書き出し中に DTD が生成されません。
- **書き出し内で DTD 生成**: 書き出し中に生成された DTD は XML ファイル内に組み込まれます (内部 DTD)。したがって、生成された XML ファイルは独立しています。
- **独立した DTD を生成**: DTD は別ファイル内に生成されます (外部 DTD)。外部 DTD は複数ユーザー間で共有できるため、さまざまなソースから生成された XML 文書の構造を統一することができます。
  **注:** 4Dでは、テーブルとフィールドに同じ名前を設定することができます。しかし、XML 言語では異なる要素に対して同じ名前を使用することが禁止されています。したがって、**DTD 生成**オプションを使用する場合は、書き出される4Dデータに同じ名前を持つテーブルとフィールドが含まれてはいけません。同じ名前が存在する場合、生成された XML ファイルは妥当ではなくなり、XML パーサーから開けなくなります。
- **DTDを参照**: 関連する**参照 (...)** ボタンを使用すると、このオプションにより既存の外部 DTD ファイルを指定することができます。4Dは、この DTD への参照を書き出しファイル内に組み込みます。

**XSL**

XSL (eXtensible Stylesheet Language) を使用すると、XML 文書中に定義された各要素の表示スタイルを指定することができます。つまり、この言語を用いて XML 文書の内容の処理や表示に使うスタイルシートを模式的に定義します。詳細は、4D *ランゲージリファレンス*マニュアルのXSL 変換の管理を参照してください。

- **既存のXSL を参照**
  このオプションを使用すると、関連する**参照**ボタンを用いて、書き出されるデータにXSLファイルを関連付けることができます。

XSL ファイルを選択した場合は、次の2つのオプションを用いて、書き出されたデータでこのファイルを利用する方法を設定しなければなりません:

- **XSL 変換を適用**: このオプションを選択すると、4Dはデータ書き出し時に、指定された XSL ドキュメント上で XSL データの変換を直接実行します。
- **XSLファイルへのリンク挿入**: このオプションを選択すると、4Dは書き出された文書内に、指定のXSL への参照を挿入します (変換は行われません)。
  したがって、書き出されたXML ファイルには、このXSL ファイルへの参照が含まれます。

**XML オプション**

- **空白を置換**: このオプションを選択すると、生成された XML ファイルの要素名フィールドにある"空白"文字が下線 ("_") で置き換えられます。
  XML の要素名フィールドでは空白文字が許可されないため、デフォルトとしてこのオプションが選択されています。
  ただし、特別な目的のために、必要に応じてこのオプションの選択を解除することができます。もちろんこの場合、生成後のファイルは、XML に関してW3C が定義した一般的な構文規則に準拠しなくなります。

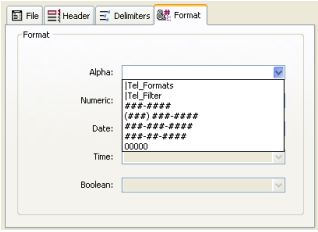
- **バイナリフィールドをBase64でエンコード**: このオプションを選択すると、4Dは書き出されるバイナリフィールド (BLOB やピクチャータイプフィールド) に"data::base64," ヘッダーを追加します。このオプションが選択されていない場合、フィールドはBase64にエンコードされますが、ヘッダーは付加されません。

- **インデントあり**: このオプションにより自動インデントが書き出されたデータに適用されます。インデントによりXML 要素の階層が視覚化されます。

- **PNGとしてピクチャーをエンコード**: このオプションを選択すると、書き出されるピクチャーは元のフォーマットにかかわらず自動で PNG フォーマットにエンコードされます。このオプションが選択されないと、ピクチャーはネイティブフォーマットにエンコードされます。
  ピクチャーが SVG で書き出される場合には、ピクチャーのプロパティを保持するために、このオプションは選択しないことをお勧めします。

#### フォーマットページ

このページでは、書き出し用フォーマットを設定することができます。このオプションは4Dフォーマットを除くすべてのファイルフォーマットで表示されます。デフォルトで4D標準フォーマットが使用されます。



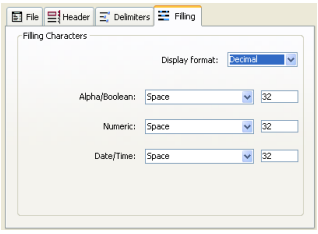

**注:** カラムのヘッダーエリアのコンテキストメニューを使用して、カラムの書き出しフォーマットを設定することも可能です (書き出すデータを指定するの手順5を参照してください)。

プレビューエリアから、文字、数字、日付、時間、ブールタイプのフィールドに対し、書き出し用フォーマットを設定することができます。プレビューエリアでカラムを選択すると、データタイプに対応するするコンボボックスが利用可能になります。コンボボックスからフォーマットを選択するか、(文字、数字、ブールデータタイプの場合は) 独自のフォーマットを直接入力します。4Dの表示フォーマットについての詳細は、表示フォーマットを参照ください。

- **ISO DateTime フォーマットについて**
  日付や時間タイプのデータの書き出しに、**ISO DateTime** 表示フォーマットを使用することができます。このフォーマットは、XML の日付と時間の表現規格 (ISO8601フォーマット) に対応しています。例えば、このフォーマットでは、"May31,2008at1:20p.m."という日付や時間は、"2008-05-31T13:20:00"と表わされます。
  4Dでは、1つのフィールド内に日付と時間の両方を保存することができません。しかし、日付や時間がXML 規格に従って保存されるように、このフォーマットでデータを書き出すことができます。日付の書き出しを行う場合、書き出された値のスタイルは"2008-05-31T00:00:00"になります。一方、時間を書き出す場合、そのスタイルは"0000-00-00T13:20:55"になります。

#### 埋め文字文字ページ

このページは、固定長テキストファイルにのみ表示されます。このページを使用し、それぞれのデータタイプに対して使用する埋め文字を定義することができます。デフォルトの埋め文字はスペース (空白) です。




固定長テキストファイルフォーマットに関する詳細は、ファイルフォーマットを参照してください。

**文字／ブール** (ブールフィールドは文字として読み込まれます)、**数字** (整数、倍長整数、実数)、**日付／時間**という3つのデータタイプに対して、異なる埋め文字を使用することができます。通常、固定長形式のフィールドを埋める場合は空白文字が用いられます。ドロップダウンリストには、それ以外の選択肢が提示されています。右側の入力エリアには、選択した埋め文字が表示されます。他の文字を使用したい場合、その文字を入力エリアに直接入力してください。

このページの上部にあるメニューを使用すると、別のフォーマットで埋め文字を表示することができます。**文字**、**10進数** (デフォルトフォーマット)、**16進数**が使用可能です。文字フォーマットを選択した場合、印刷不可の埋め文字は表示されません。

### フォームを使用してデータを書き出す

フォームを使用して書き出しを行いたい場合、書き出しダイアログボックス上部にある**フォーム**タブをクリックします。この後、選択したテーブルのフォーム一覧からフォームを選択することができます。

フォームを使用する主な利点は、各レコードが書き出される際に、フィールドや変数に関連付けられたフォームメソッドとオブジェクトメソッドが実行されることです。これにより、書き出し中にデータ処理を行うことができます。フィールドや変数は、フォームの入力順に書き出されます。ただし、これらのメソッドの結果はプレビューエリアに表示されない点に注意してください。

フォームにはボタンを置かないでください。また、サブフォームオブジェクトは無視されます。

**4DServer:** サーバー上のストアードプロシージャのメソッドを含むフォームを使用して、データを書き出すことはできません。つまり、各イベントはサーバーにより管理されないため、割り当てたメソッドが呼び出されません。

## ODBC データソースを使用した読み込みと書き出し

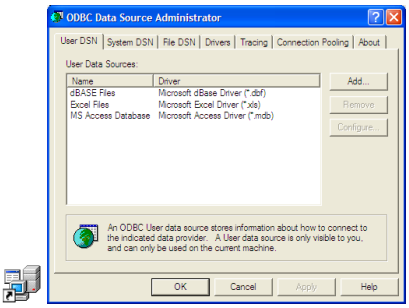
4Dでは、ODBC データソースからのデータ読み込みや、データソースへのデータ書き出しを行うことができます。

ODBC プロトコル (Open DataBase Connectivity) は、さまざまなアプリケーションが SQL ランゲージを用いて互いに通信を行うためのインタフェースです。ODBC データソースは、アプリケーションが開かれていない場合でも、そのアプリケーションに属すドキュメントに対して常に開かれた"窓"のようなものです。多くのアプリケーション (データベース、スプレッドシート、データマネージャなど) では、特定の ODBC ドライバを使用して、データをODBC ソースとして公開することができます。

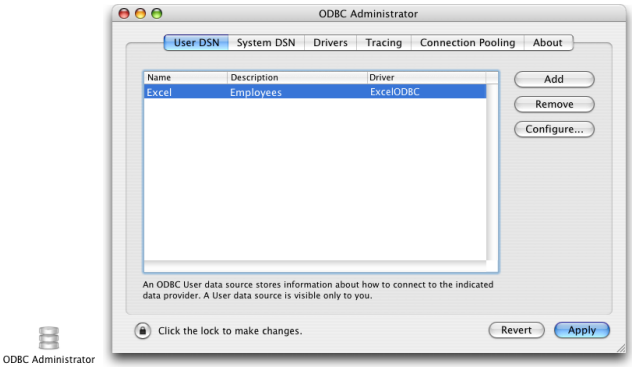
ODBC プロトコルをサポートするあらゆるアプリケーションは、ODBC ソースのデータの読み込みや書き出しを行えます。4Dの読み込み機能を使用すると、ODBCソースのデータを取得することができます。また、4Dの書き出し機能により、データを ODBC ソースへ挿入することができます。

**ODBC ソースを定義する**

Windows において、ODBC ソースの定義は**管理用ツール**を使用して、**データソース (ODBC)** 設定パネルで行います。




Mac OS X において、ODBC ソースの定義は**ユーティリティ**フォルダ内にある **ODBC アドミニストレータ**プログラムを使用して行います。




これらのダイアログボックスに関する詳細は、OS のドキュメントを参照してください。

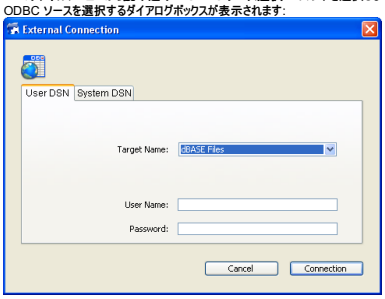
**ODBC ソースからデータを読み込む**

ODBC ソースからのデータ読み込みに関する概念は、他のタイプのデータを読み込む場合と同じです (ファイルからデータを読み込む参照)。ただし、ODBC ソースを選択するために中間的なダイアログボックスを使用する点が異なります。

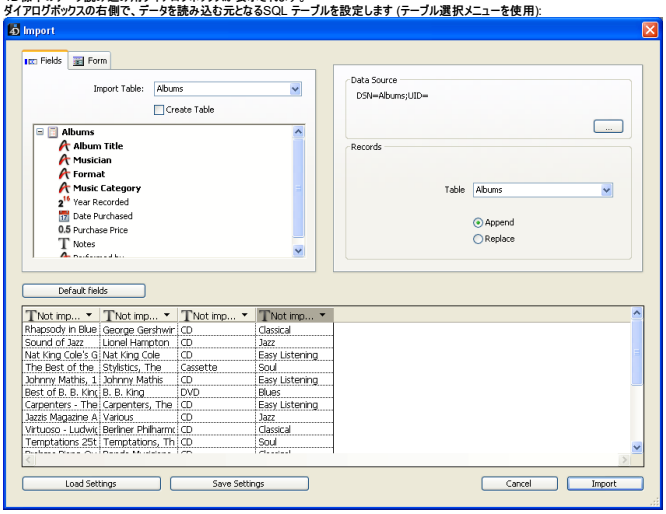
アプリケーションモードにおいて読み込みを行う場合は、IMPORT ODBCコマンドを使います。デザインモードにおいては、4Dのファイルメニューを使います。

ODBC ソースからデータを読み込むには、次の手順に従ってください:

1. 4Dの**ファイル**メニューから**読み込み > ODBC ソース選択 ...**コマンドを選択します。
   ODBC ソースを選択するダイアログボックスが表示されます:




2. 使用するデータソースと (必要であれば) アクセスID を選択し、**接続**をクリックします。
   4D標準のデータ読み込み用ダイアログボックスが表示されます。
3. ダイアログボックスの右側で、データを読み込む元となるSQL テーブルを設定します (テーブル選択メニューを使用):



4. データの読み込み先となるテーブルと4Dフィールドを選択するか、または**テーブル作成**オプションをチェックします (コンパイル済データベースには、このオプションを使用できません)。
   他のすべてのタイプの読み込みと同様に、プレビューエリアには読み込まれるデータをフィールドに割り当てた状態が表示されます。また、**フォーム**タブをクリックし、フォームを用いてデータを読み込むこともできます。これらの機能に関する詳細は、ファイルからデータを読み込むを参照してください。
5. ダイアログボックスの右側にある読み込みオプションを選択します。
   - **データソース**: このエリアには、データベースが接続されるデータソースの記述が表示されます。**参照...**ボタンをクリックすると、データソースを選択するダイアログボックスを再び表示することができます。
   - **レコード**: ソースから読み込むデータを読み込み先テーブルのレコードに追加したり、または置き換えることができます。
6. **読み込み**ボタンをクリックしてデータを読み込みます。
   データの読み込み後、4Dとデータソースとの接続は自動的にクローズされます。

## 読み込み設定と書き出し設定の保存とロード

読み込みおよび書き出しダイアログボックスの各設定をディスクへ保存したり、またはディスクから読み込むことができます。

ファイルや ODBC ソースの読み込みや書き出しダイアログボックスを使用して、この操作を実行することができます。ただし、ODBC の読み込み／書き出し用のダイアログボックスに保存されたプロジェクトは、標準の読み込み／書き出し用ダイアログボックスとの互換性がありません (逆の場合も同じ)。

読み込みや書き出し設定を保存またはロードするには、コマンドエリアの**設定保存**ボタン、または設定読み込みボタンをクリックします:

読み込みと書き出しの設定ファイルには、Windows では".4SI" (Mac OS でのファイルタイプは"4DSI") という拡張子が付加されます。設定ファイルには、読み込みと書き出しダイアログボックスで指定したすべての設定が保存されます:

- ファイルやODBC ソースの名前とアクセスパス
- 選択されたテーブルとフィールド、および書き出しに使用されたフォーム
- 読み込みオプションと書き出しオプション (ファイルタイプ、区切り文字など)

この機能を使用して、読み込みや書き出し処理を自動化することができます。IMPORT ODBCコマンドやEXPORT ODBCコマンドと一緒に、この設定ファイルを使用すると特に有効です。

# Web サービスの公開と使用


はじめに

4Dで Web サービスを公開する

4Dから Web サービスへサブスクライブする

## はじめに

### Web サービスとは？

Web サービスは、一連の機能をエンティティ (実体) として集めたものであり、ネットワーク上に公開されます。これらの機能は、Web サービスに対応し、かつ同じネットワークに接続する任意のアプリケーションから呼び出したり、使用することができます。当然ながら、Web サービスはインターネット上に公開された上で、最大限に利用されることを目的としています。

Web サービスにより、運送会社の貨物配送管理や eコマース、市場価格の監視など、あらゆる種類のタスクを実行することができます。

このサービスを公開するプログラムは、"サーバー"と呼ばれます。したがって、Web サービスに対応しているアプリケーションはすべて、これらの機能を一つ以上利用することができます。これが"クライアント"プログラムです。Web サービスの利点は、さまざまな情報システムとの相互運用が可能であるところです。つまり、このシステムが機能するために、サーバー側とクライアント側のプログラムが互いに互換性を持つ必要はありません。クライアントアプリケーション側からみると、Web サービスは"ブラックボックス"のようなものです。これは、ある値を Web サービスに送信すると、処理結果である別の値が返されるためです。

サーバーが提供する Web サービスは、パブリックまたはプライベートのいずれのタイプでも構いません。インターネット上にはパブリックタイプの Web サービスが多数提供されており、任意のアプリケーションが無料で要求を送信することができます。

Web サービスは、W3C (World Wide Web Consortium、インターネットに関する管轄機関) やコンピュータ産業の主要企業により管理されているため、信頼性が高く永続的で機能向上可能なコネクティビティソリューションの代名詞となっています。

### Web サービスの運用 – 主な定義

Web サービスは、原則的に HTTP 通信プロトコルを使用して通信を行います。

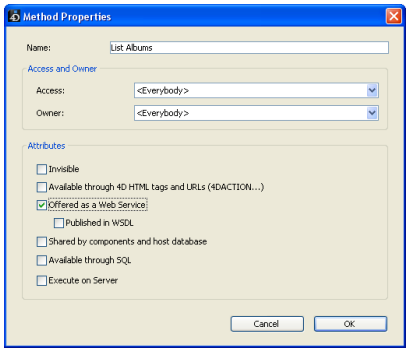
- **SOAP**: Web サービスは、"オープン"なハイレベル通信プロトコルである SOAP (Simple Object Access Protocol) を使用します。このプロトコルは、メッセージ構造 (エンベロープ) レベルと交換データレベルの双方において、全面的に XML 言語を使用しています。このプロトコルの運用は、RFC (Request for Comment : インターネットに関するさまざまな局面を標準化する文書) により定義され、広範囲にわたり互換することが保証されています。
  Web サービスの運用原理は、次の通りです。まず、Web サービスのクライアントが SOAP プロトコルを介して、リクエストを XML でサーバーに送信します。サーバーはこのリクエストを解析し、要求された処理を実行した後、同じプロトコルと言語を使用してその応答を返します。
- **WSDL**: 通常、Web サービスのサーバーは、提供するサービスの利用に関する仕様を定義するため、WSDL (Web Service Description Language) を公開します。この WSDL により、Web サービスのサーバーは、提供するサービスの"操作説明"を公開することができます (URL、メソッド一覧、パラメーター等)。これは XML ファイル形式で記述され、一般的にサーバーアプリケーション自体が作成します。このファイルは、必須ではありません。
- **UDDI**: UDDI (Universal Description Discovery and Integration) は、すべてのパブリック Web サービスが登録されている世界規模のデータベースです。ただし、必ずしも Web サービスをパブリックにする必要はなく、大抵の場合は、この登録が不要である点に注意してください。

### Web サービスを4Dへ統合する

4Dは、Web サービスのサーバーやクライアントとして使用することができます。Web サービスを4Dへ統合する方法はシンプルで安全です。いくつかの設定を行うだけで、公開やサブスクライブの状況を正確に監視することができます。

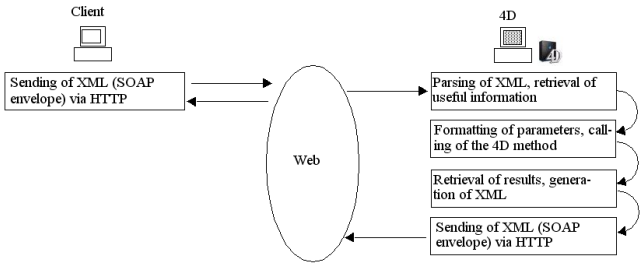
#### Web サービスサーバーとしての4D

特に大きな変更を加えなくても、任意のプロジェクトメソッドを Web サービスとして公開することができます。公開はメソッドのプロパティで行います:




**注:** 4Dを使用して Web サービスの公開を行うには、特定のライセンス (SOAP ライセンス) が必要です。

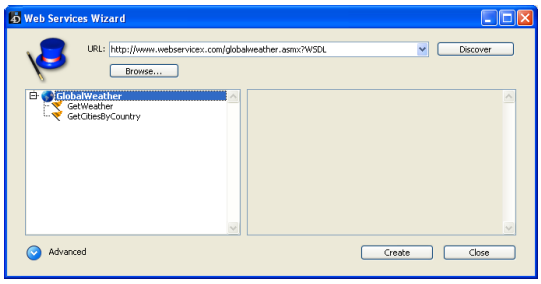
4D Web サーバーは、サービスの管理、およびWSDL ファイルの公開や保守を自動的に処理します。XML でのリクエスト内容の解析やパラメーターのフォーマット、結果の送信などの操作は4Dが実行し、特定のプログラムは必要ありません。




しかし、リクエスト処理をカスタマイズしたい場合は、特定の4Dランゲージコマンドを使用することができます (4D *ランゲージリファレンス*マニュアルのWeb サービス (サーバ)参照)。

#### Web サービスクライアントとしての4D

インターネット上やネットワーク上で提供されるあらゆるタイプの Web サービスをデータベースから利用することができます。大抵の場合は、Web サービスウィザードを使用すると、ほとんどプログラムを行わなくても、すべての Web サービスを直ちに使用することができます:




4Dでは、ネットワーク経由でリクエストを送信し、応答を取得することにより Web サービスを使用します。これらの処理は、"プロキシ"メソッドが担当します。Web サービスを呼びだすプロキシメソッドはすべて自動的に作成され、プログラミングなしで実行されます。つまり、自分のコードからこれらのメソッドを呼び出すだけです。

4Dのランゲージコマンドを使用し、サーバー側と同じ方法でこれらのメソッドをカスタマイズすることができます。(Webサービス (クライアント)参照)

#### Web サービスのセキュリティ

4Dが公開する Web サービスは、4D Web サーバーに対して設定されたセキュリティメカニズムを継承します。したがって、Web サービスのリクエストでは、パスワード、On Web AuthenticationとOn Web Connection の各データベースメソッド、SSL プロトコルの使用など、今までの Web リクエストと

同じ設定を利用することができます。

さらに、特定の設定を用いて (例えば、SOAP Get info関数やSOAP request関数)、Web サービスの公開を詳細に制御することもできます。

クライアント側では、Web サービスのサーバーへの接続はSSL を用いた暗号化モードで実行できます。ここでもWEB SERVICE AUTHENTICATEコマンドを用いて、認証を必要とするサーバーへの接続を行うことができます。

#### RPC、DOC、複合型の互換性

Web サービスのコミュニケーションレイヤ (転送、サービスのコール、交換セキュリティを保証する) は、**RPC** (Remote Procedure Call) モードと **DOC** (Message/Document) モードという2種類のモードで運用することができます。この2つのモードは、サーバーとクライアントに対するリクエストや応答を作成するレベルにおいて異なります。

4Dは、RPC モードと DOC モードをサポートします:

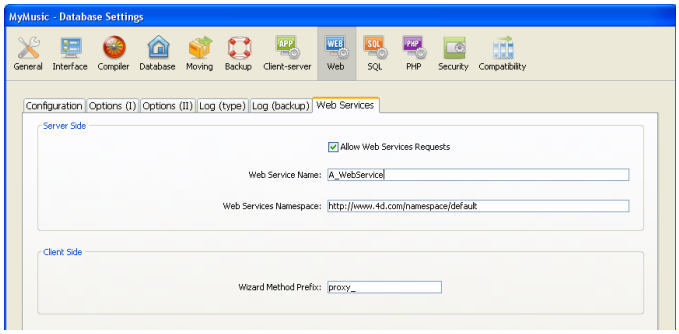
- クライアント側では、Web サービスウィザードにより透過的にサポートされます。生成されるコードは、自動的に公開モードに合わせられます。
- サーバー側でも、透過的にサポートされます。メソッドは RPC モードと DOC モードで自動的に Web サービスとして公開されます。このモードの選択は、WSDL の URL を通して行われます (WSDL ファイルを生成する参照)。
  - RPC モードで4DのWeb サービスを公開するには、WSDL の次のURL を使用します:
    **http://ServerAddress/4DWSDL**
    4Dのサーバーは、リクエストと応答を RPC モードで処理します。RPC モードのリクエストは自動的に次のアドレスへ送信されます:
    http://ServerAddress /4DSOAP
  - DOC モードで4Dの Web サービスを公開するには、WSDL の次の URL を使用します:
    **http://ServerAddress/4DWSDL/DOC**
    4Dのサーバーは、リクエストと応答を DOC モードで処理します。DOCモードのリクエストは自動的に次のアドレスへ送信されます:
    http://ServerAddress /4DSOAP/DOC

SOAP プロトコルにより、**単純型** (simple types) と**複合型** (complex types) という2種類の XML データのやり取りが行われます。RPC モードで公開されたWeb サービスのデータは、いずれのタイプでも構いません。これとは逆に、DOC モードで公開された Web サービスのデータは、意図的に複合型となります。4Dは、単純型と複合型のデータを使用して Web サービスをサポートします。

4Dデータベースでは複合型の XML データを直接使用することができず、特別な処理が必要になります。ほとんどの場合は、Web サービスウィザードがユーザの代わりにこの処理を実行しますが、4Dの XML ランゲージコマンドを用いてこの処理を完了しなければならない場合もあります。この件に関する詳細は、Processing complex typesを参照してください。

#### 環境設定

データベースの環境設定 ("Web"テーマ) の **Webサービス**ページを使用して、Web サービスの公開や利用に関する一般的なパラメーターを定義することができます:




これらのパラメーターの働きについては、後章で説明します。各オプションに関する詳細は、Web/Web サービスページを参照してください。

## 4Dで Web サービスを公開する

通常、4Dでは、次の3つの段階を経て Web サービスが公開されます:

1. 公開されるメソッドの作成
2. 公開に関する設定 (WSDL)
3. 公開

さらに、カスタマイズを行う段階もありますが、これは必須ではありません。

### Web サービスメソッドを作成する

Web サービスとして公開する目的で、あらゆるタイプのプロジェクトメソッドを作成することができます。このメソッドは、引数を受け取り、結果を返さなくてはなりません。また、その引数は、コンパイラテーマのコマンドを使用して、必ずメソッドヘッダーで宣言しなくてはなりません。

デフォルトとして、4Dは Web サービスとして公開される際に、メソッドの処理に必要となる引数をフォーマットします。しかし、SOAP DECLARATIONコマンドを使用すると、これらの引数を変更することができます。

4Dは、SOAP 経由で送受信したデータの暗号化と解読を自動的に処理します。

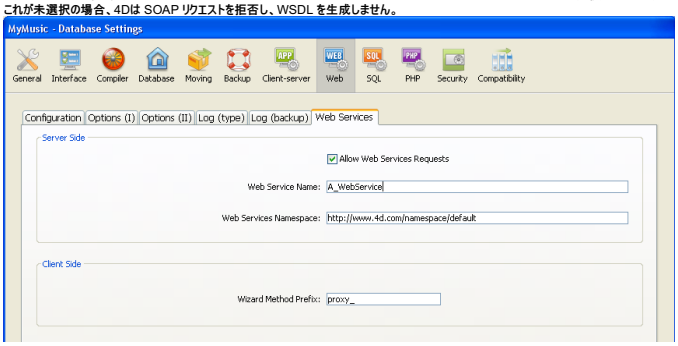
**警告:** SOAP リクエストでは、メソッド名が XML タグとして使用されます。タグ名に関する XML 規格に準拠して、Web サービスとして公開されるメソッド名にはスペースや拡張文字を使用しないでください。また、数字を頭文字として使用できません。次の Latin 文字だけを使用することができます: ([AZaz0-9._] | '-') *

Web サービスとして公開されるメソッド開発の定義や監視を行うには、"Web サービス"テーマのコマンドを使用しなくてはなりません。これらのコマンドについては、4D ランゲージ/ファレンスマニュアルのWebサービス (サーバ)を参照してください。

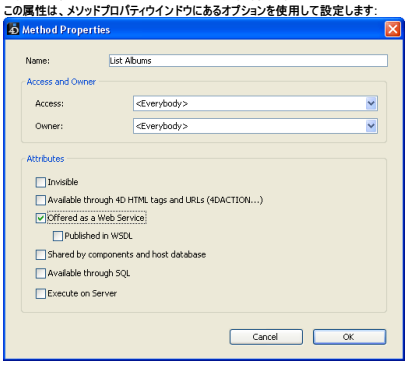
### メソッドの公開

データベースの1つ以上のメソッドを Web サービスとして公開するためには、次の4つの条件を満たしていなくてはなりません。

- SOAP サーバーとして使用されるマシンに (4D Server、シングルユーザ版4D) 4Dの SOAP ライセンスが登録されている。
- 4D Web サーバーが起動している。
- アプリケーションの環境設定ダイアログボックスにおいて、SOAP ページの **Web サービスリクエストを許可する**オプションが選択されている。
  これが未選択の場合、4Dは SOAP リクエストを拒否し、WSDL を生成しません。




  このオプションが選択されていると、4Dは WSDL ファイルを作成します (WSDL ファイルを生成する参照)。
- 公開される各メソッドに、**Web サービスとして提供**属性が設定されている。
  この属性は、メソッドプロパティウインドウにあるオプションを使用して設定します:




  **注:** このウインドウに関する詳細は、プロジェクトメソッドプロパティを参照してください。
  **Web サービスとして提供**オプションが選択されている場合、対応するメソッドは SOAP リクエストにより Web サービスとして呼び出されます。
  **注: WSDL で公開する**オプションも選択されている場合、そのメソッドはサーバーの WSDL 上に表示されます (次の節を参照)。

4Dエクスプローラのメソッドページにおいて、Web サービスとして提供されるメソッドは特定のアイコンで表わされます (WSDL ファイルに公開されるメソッドも同様: )。

### WSDL ファイルを生成する

WSDL は、4Dメソッドの呼び出しに必要となるシンタックスと情報 (メソッド名、URL、引数等) を XML 言語で記述します。

4Dにおいて、WSDL は単独の Web サービスに相当します。WSDL はメソッドとその引数を定義し、特定の場所で参照されます。4D内で、WSDL は実在上の"ファイル"ではありません (メモリ上だけ に存在し、ディスクには書き込まれません)。つまり、WSDL は URL であり、RPC モードで公開される Web サービスの場合は **4DWSDL**という名前が付けられ、DOC モードで公開される Web サービスの場合は **4DWSDL/DOC** という名前が付けられます。また、常に Web サーバーのルートに置かれます。

例えば、Web サービスのアドレスが http://www.myserver.com である場合、次の URL で WSDL を参照することができます:

- http://www.myserver.com/4DWSDL (RPCモード)
  http://www.myserver.com/4DWSDL/DOC (DOCモード)

WSDL 内のメソッドを追加、または削除するには、メソッドプロパティダイアログボックスの該当するオプションを選択するか、または選択解除します。すると、4Dは即座に WSDL の内容を更新します。

**注:** 4Dのエクスプローラにおいて、WSDL で公開されるメソッドは特定のアイコンにより区別されます。

Web サービスリクエストが許可されると、メソッドプロパティウインドウで少なくとも1つのメソッドに **WSDL で公開する**オプションが選択されていれば、4Dは 4D Web サーバーの WSDL を動的に自動生成します。デフォルトとして、このオプションは選択されていません。

### Web サービス名をカスタマイズする

インターネット上に公開される Web サービスにはそれぞれ名前が付けられています。この名前を使用して、SOAP サーバーレベル (サーバーが複数の Web サービスを公開する場合) および Web サービスディレクトリ上の双方のサービスを区別します。

デフォルトとして、4Dは"*A_WebService*"という名前を使用します。このパラメーターは、データベースの環境設定において Web テーマの Web サービスページで変更することができます。

**警告:** タグ名に関する XML 規格に準拠して、文字列にはスペースや拡張文字を使用しないでください。また、数字を頭文字として使用できません。次の Latin 文字だけを使用することができます: ([A-Za-z0-9._] | '-') *

### 名前空間をカスタマイズする

インターネット上に公開される Web サービスは、それぞれがユニークでなければなりません。XML の名前空間を使用すると、Web サービス名の一意性が保証されます。名前空間は任意の文字列であり、一連の XML タグを独自の方法で識別するために使用されます。一般的に、名前空間はその会社の URL で始まります (http://mycompany.com/mynamespace)。この場合、特別な URL を指定しなければならないという訳ではなく、使用する文字列が一意であることが重要になります。

デフォルトとして、4Dは **http://www.4d.com/namespace/default** という名前空間を使用します。このパラメーターは、データベースの環境設定においてWebテーマの"Webサービス"ページで変更することができます。

### 公開メソッドへコメントを追加する

Web サービスとして提供され、WSDL で公開されるメソッドに割り当てたコメントは、"documentation"フィールドとして自動的に WSDL ファイル内に現われます。

これらのコメントはエクスプローラーのメソッドページにて設定されている必要があります (コメントを使用する参照)。

このメカニズムは、公開されるメソッドの説明や文書化を行うために使用されます。このフィールドの解釈や処理方法は、クライアントの Web サービスの実装により異なります。

**4Dにより公開された Web サービスへアクセスする**

4Dにより Web サービスが公開されると、Web サービスをサポートするあらゆるクライアントアプリケーションからその Web サービスに接続することができます。アクセスモード、ならびに Web サービスのサーバーとやり取りする情報の処理方法は、その操作に使われるクライアントアプリケーション側で決まります。

4D Web サービスを使用する上で必要となる情報はすべて (サービスの URL、使用されるパラメーター等)、4Dの WSDL に公開されます。原則として、Web サービスを使用する際は、この情報を取得するために、まず SOAP サーバーの WSDL を読み込むことから始めてください。4Dにおいて、WSDL のURL は、**http://サーバーアドレス/4DWSDL** (RPC モード) または、**http://サーバーアドレス/4DWSDL/DOC** (DOC モード) です。

ただし、このステップは必須ではありません。SOAP サーバーへはダイレクトに接続することができます。

SOAP リクエストの作成に必要となる値の一覧、およびその定義方法を次に示します:

- **4Dにより公開された Web サービスへのアクセス URL**
  http://サーバーアドレス/4DSOAP/ (RPC モード) または
  http://サーバーアドレス/4DSOAP/DOC (DOC モード)
  (カスタマイズ不可)

- **Web サービス名**
  デフォルト: A_WebService
  値のカスタマイズ可能 (Web サービス名をカスタマイズする参照)。

- **公開メソッド名**
  開発者が定義した4Dプロジェクトメソッド名 (Web サービスメソッドを作成する参照)。

- **メソッドの引数**
  引数はメソッド内で宣言されていなければなりません (開発者が定義)。
  デフォルトの SOAP 名: FourD_arg0、FourD_arg1... FourD_argn
  SOAP DECLARATIONコマンドで名前をカスタマイズ可能。

- **名前空間**
  デフォルト: http://www.4d.com/namespace/default
  値のカスタマイズ可能 (名前空間をカスタマイズする参照)

- **SOAP Actionフィールドの内容**
  サービス名#メソッド名 (カスタマイズ不可)。

## 4Dから Web サービスへサブスクライブする

4Dを使用して、Web サービスへサブスクライブすることができます。言い換えれば、自分のデータベース内から外部の Web サービスを呼び出すことができます。

インターネット上で利用可能な Web サービスを使用すると、株価情報へのアクセスや貨物配送の追跡、複雑な計算の実行など、数々の追加機能を容易にデータベースへ組み込むことができます。インターネット上で利用できる多数の Web サービスが、ほとんどすべての要求を満たしてくれます。

また、別のデータベースで自分が公開した Web サービスへサブスクライブすることも可能です。このような方法により、さまざまな4Dデータベースが各々の間で通信できるようになります。

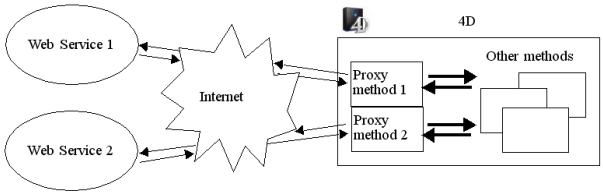
**原則**

あらゆる4Dデータベースは、インターネットに接続するだけで Web サービスへサブスクライブすることができます。

通常、Web サービスを呼び出すには、次に説明する手順を踏まなければなりません:

1. サブスクライブしようとする Web サービスの URL を取得します。
   URL を取得するには、インターネット上に公開された Web サービスの一覧が登録されている Web サイトや (例えば、www.xmethods.net)、UDDI のようなディレクトリを利用することができます。大抵の場合、その Web サービス用の WSDL ファイルのURL を取得しなければなりません。
   注: 4Dは、RPC モードや DOC モードで公開された Web サービスを使用することができます (RPC、DOC、複合型の互換性参照)。
2. Web サービスウィザードを使用して、利用する Web サービスの WSDL の内容を解析し、対応するプロキシメソッドを生成します。プロキシメソッドはローカルなプロジェクトメソッドで、Web サービスへの問い合せや戻り値の取得を行います。この手順については、後述の Web サービスウィザードを使用するの節を参照してください。
   **注:**
   - WSDL ファイルを使用しなくても、Web サービスウィザードでプロキシメソッドを作成することができます (使用するパラメーターを手動で入力するだけです)。
   - また、Web サービスウィザードを使用しなくても、メソッドエディターでプロキシメソッドを作成することもできます (上級ユーザ向け)。
3. データベースのコード内で適切な引数を渡して、必要になるたびにプロキシメソッドをコールします。
   この手順については、後述のプロキシメソッドを呼び出すで説明します。

プロキシメソッドは、Web サービスへの接続を行います:



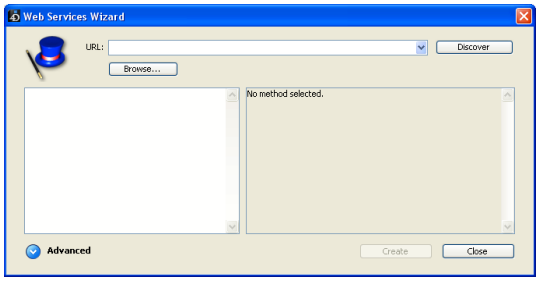

**Web サービスウィザードを使用する**

4Dアプリケーションからの Web サービスへのサブスクライブは、Web サービスウィザードがすべて処理します。このウィザードは次の処理を自動的に実行します:

- 利用する Web サービスの WSDL ファイルを解析。
- 作成するプロキシメソッドのパラメーターを定義。
- プロキシメソッドを作成。

**ウィザードウインドウ**

Web サービスウィザードのウインドウを開くには、4Dの**デザイン**メニューから **Web サービスウィザード...**を選択します。ウィザードウインドウが表示されます:




**注:** エクスプローラのメソッドページにあるオプションメニューから Web サービスウィザードを表示することもできます (参照)。

このウインドウには3つのエリアがあります:

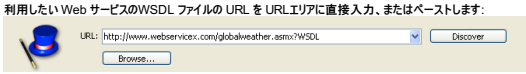
- "URL:"エリアを使用すると、選択した Web サービス用の WSDL ファイルの URL を入力、または選択することができます。このエリアはコンボボックスになっており、以前に入力された値がドロップダウンリスト形式で保存されています。
- 中央のエリアには、サービス名や公開メソッドなど、WSDL ファイルの内容を解析した結果が表示されます。
- 下側のエリア ("詳細設定"パラメーター、デフォルトでは非表示) には、中央のエリアで選択したメソッドの引数が表示されます。

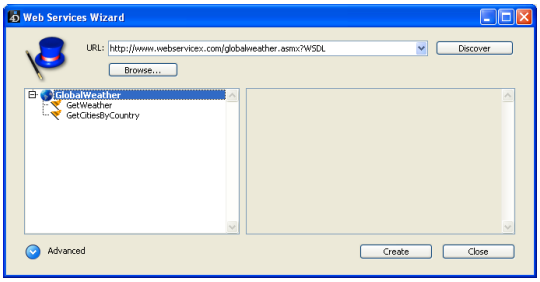
**検索** ボタンにより、指定した WSDL ファイルの解析が開始され、その結果が情報エリアに表示されます。
**ブラウズ...**ボタンにより、標準のファイルオープン用ダイアログボックスが表示され、ローカル上に保存された WSDL ファイルを選択することができます。すると、URLエリアには"file://"で始まるファイルへのアクセスパスが表示されます (このエリアにアクセスパスを手動で入力することもできます)。
**作成**ボタンを使用すると、選択した Web サービスに対応するプロキシメソッドが生成されます。
**閉じる**ボタンを使用すると、Web サービスウィザードダイアログボックスが再度閉じられます。

**WSDL の解析とプロキシメソッドの作成 (標準モード)**

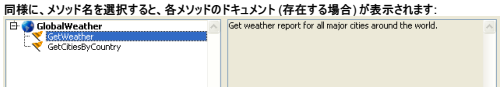
Web サービスウィザードの主な使用目的は、WSDL ファイルを解析して、対応するプロキシメソッドを生成することです。この標準的な処理はすべて自動的に行われ、ユーザー側のプログラミングや特別な知識を必要としません。

WSDL ファイルを解析し、プロキシメソッドを生成するには、次の手順に従ってください:

1. 利用したい Web サービスのWSDL ファイルの URL を URLエリアに直接入力、またはペーストします:
   
   
   例えば、この URL は、Web サービスの"ディレクトリ"から参照するか、あるいは Web サービスのサーバーにより直接通信されたものを使用できます。
   また、ローカルな URL、つまり自分のハードディスク上に保存された WSDL ファイルのアドレスを指定することもできます。これを行うには、ブラウズ...ボタンをクリックして、ローカル上の WSDL ファイルを選択するか、または直接"URL:"エリアにファイルのアクセスパスを入力します。ローカルファイルのアクセスパスは"file://"で始まり、その後に標準システムのフォルダ区切り文字を指定します。必ず絶対パスを渡さなければなりません。
2. **検索**ボタンをクリックして、4Dに WSDL ファイルの内容を解析させます。
   しばらくすると、中央のエリアにファイルの解析結果が表示されます。Web サービスの名前ならびに公開されたメソッドが階層リスト形式で示されます。

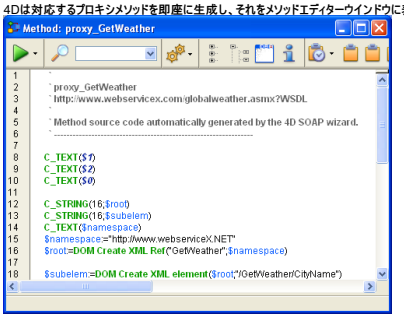



**注: Shift** キーを押しながら**検索**ボタンをクリックすると、WSDL ファイルの XMLソースコードをデフォルトの Web ブラウザに直接表示することができます。
Web サービス名をクリックすると、そのドキュメント (存在する場合) がウインドウの右側エリアに表示されます。ドキュメントがない場合は、"ドキュメントがありません"というメッセージが現われます。
同様に、メソッド名を選択すると、各メソッドのドキュメント (存在する場合) が表示されます:




**注:** WSDL ファイルを解析した結果、複合型のパラメーターが存在することが判明した場合に、ウィザードでは関係するメソッドの横に黄色の旗が表示されます。

3. 利用したい Web サービスメソッドを選択し、**作成**ボタンをクリックします。
4Dは対応するプロキシメソッドを即座に生成し、それをメソッドエディターウインドウに表示します:




プロキシメソッドの名前は、デフォルトの接頭辞である"proxy_"とWeb サービスメソッド名とを連結して設定されます。デフォルトの接頭辞は、データベースの環境設定の[Web/Web サービスページ]で変更することができます。また、作成後にプロキシメソッドの名前を変更することもできます。名前を変更しても、そのメソッドの動作には影響しません。

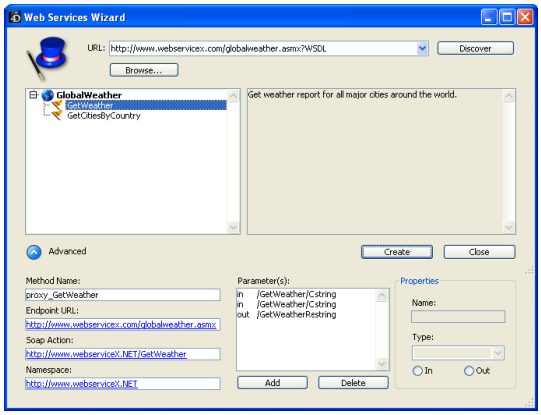
#### 詳細設定パラメーターを使用する

WSDL ファイルの解析に基づいて Web サービスウィザードで生成されたプロキシメソッドは、即座に実行可能であり、そのまま使用できます (標準モード)。

しかし、WSDL の解析により得られたパラメーターを変更したい場合もあります。例えば、プロキシメソッド名を変更することができます。

また、Web サービスウィザードを使用して、自分でパラメーターを入力したプロキシメソッドを作成することもできます。この場合は、WSDL パーサーを使用しないでください。
メソッドを作成するために、すべてのパラメーターを入力する必要はありません。
パラメーターをひとつも入力せずに、プロキシメソッドの"テンプレート"を作成することも可能です。後から4Dのプログラムを使用して、この"テンプレート"にパラメーターを設定することができます。

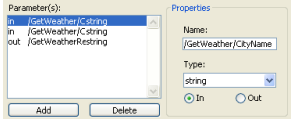
これらの非標準モードでは、Web サービスウィザードの詳細設定パラメーターを使用しなければなりません。このパラメーターを表示するには、ウィザードウインドウの左下にある展開用のボタンをクリックします。すると、詳細設定パラメーターの各フィールドが表示されます。メソッドが選択されている場合は、これらのフィールドに現在のパラメーターが表示されます:




パラメーターはすべて変更可能です。ただし、WSDL の解析により生成されたパラメーターを変更すると (メソッド名を除く)、結果として Web サービスの動作が変わる可能性があるため、注意深く行わねばなりません。

詳細設定パラメーターについて次に説明します:

- **メソッド名**: 作成されるプロキシメソッドに対してウィザードが割り当てる名前。デフォルトとして、この名前は接頭辞"proxy_" (環境設定で変更可) に続けて、選択したメソッド名を指定して設定されます。この名前は自由に変更することが可能であり (例えば、データベースに同じ名前が既に存在する場合など)、変更しても Web サービスの動作には影響を与えません。
- **Endpoint (エンドポイント) URL**: プロキシメソッドが SOAP リクエストを送信する URL。
- **SOAP アクション**: SOAPActionフィールドの内容。通常、このフィールドには"サービス名#メソッド名"という値が納められます。
- **名前空間:** Web サービスの名前空間 (詳細については、名前空間をカスタマイズするを参照)。
- **パラメーターテーブル**: この表には、公開メソッドについてパラメーターの一覧が表示されます。




表の各行にパラメーターが示されます:
  - 最初のカラムは、そのパラメーターが入力タイプ ("in") か出力タイプ ("out") かを示します。この特性は、公開メソッド側ではなくプロキシメソッド側の視点から判断されます。
  - 2番目のカラムは、パラメーター名を表わします。
  - 3番目のカラムは、パラメーターの SOAP タイプを示します。4Dにより受け入れられる各種 SOAP タイプは、プロパティエリアにあるタイプメニューに表示されます。Web サービスウィザードは、プロキシメソッド内で各 SOAP タイプを対応する4Dのタイプに割り当てます。
  次の表は、受け入れられる SOAP タイプの値と、それに対応する4Dのタイプを示しています:

| SOAP タイプ | 対応する4Dタイプ |
|---|---|
| boolean | ブール |
| int | 倍長整数 |
| time | 時間 |
| float | 実数 |
| double | 実数 |
| date | 日付 |
| string | テキスト |
| base64Binary | BLOB |
| ArrayOfBoolean | ブール配列 |
| ArrayOfInt | 倍長整数配列 |
| ArrayOfTime | 時間配列 |
| ArrayOfFloat | 実数配列 |
| ArrayOfDate | 日付配列 |
| ArrayOfString | テキスト配列 |
| AsXML | BLOB |

**注:** 厳密に言うと、AsXML タイプは SOAP タイプではなく、複合型の XML をサポートするために使用されます (複合型の処理参照)。
プロパティエリアには、この表で選択したパラメーターの特性が表示されます。Web サービスウィザードを使用すると、例えば、指定した WSDL ファイルが最新ではない
場合などに、既存のパラメーターの変更や新規パラメーターの追加を行うことができます。

- パラメーターを変更するには、パラメーターを選択してからプロパティエリアで変更を行います。
- パラメーターを追加するには、**追加**ボタンをクリックしてから、プロパティエリアでその特性を定義します。
- パラメーターを削除するには、リストからパラメーターを選択し、**削除**ボタンをクリックします。

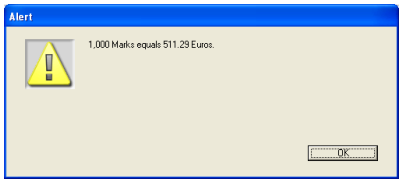
**注:** **作成**ボタンを使用して実際にプロキシメソッドが作成された場合にのみ、詳細設定パラメーターへの変更が反映されます。

**プロキシメソッドを呼び出す**

コード内でプロキシメソッドを呼び出すには、プロキシメソッド名を記述して必要なパラメーターを渡します。これらのパラメーターは、Web サービスウィザードによりプロキシメソッドのヘッダーエリアで宣言されます。4Dにおけるメソッド間の引数受け渡しに関する標準的なシンタックスに従い、パラメーターには$0、$1、$2などの名前が設定されます。これらパラメーターは、公開メソッドの詳細設定パラメーターの説明エリアに表示され (詳細設定パラメーターを使用する参照)、またそのドキュメント内で説明される場合もあります。

例えば、メソッド名"WS_EuroConverter"の例では、プロキシメソッドは次の方法で呼び出されています:

このメソッドを実行すると、次の警告が表示されます:




**複合型の処理**

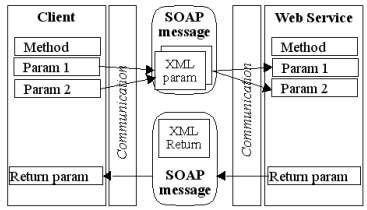
4Dでは、RPC モードや DOC モード、および複合型で公開された Web サービスを使用することができます (RPC、DOC、複合型の互換性参照)。

**注:** 実際には複合 XML タイプであるにも関わらず、データ配列は4Dにより単純型として処理されます。

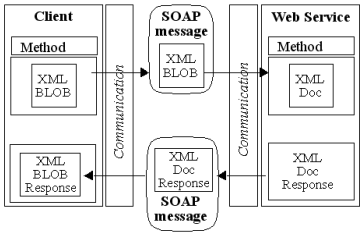
Web サービスウィザードにより生成された、複合型を含むプロキシメソッド (つまり複合型を伴う RPC モード、または DOC モードで公開されたメソッド) は、標準的なプロキシメソッドと似ています。しかし、これらの Web サービスを用いた特定のケースにおいて、WEB SERVICE CALLコマンドでは"manual"という語句を含む定数が引数として使われていることがわかります。

実際には、このような Web サービスを使用する場合は、さらに処理が必要となります。この主な理由は、複合型のやり取りが文書や XML 要素の形で行われるためです。つまり、これらの SOAP パラメーターで情報の取得や組み込みを行うには、事前に XML を解析する必要があります。これに対して単純型の場合は、パラメーターの値を直接読み取ることができます。

- *単純型の使用 (RPC)*




- *複合型の使用 (DOC)*




Web サービスウィザードでは、配列や複合型データのうち、1レベルのもの (SOAP リクエストにおける単独の階層レベル) しか完全にサポートされません。これ以上複雑な要素がリクエスト中に存在する場合、ウィザードはメソッド名の横に旗を表示して、そのことを表わします。通常、このタイプの Web サービスをサポートするには、開発者による独自の処理が必要です。

4Dでは、複合型のパラメーター (配列を除く) は BLOB 形式で処理されます。これらの BLOB の内容を処理するために、4Dの XML コマンドを使用することができます。詳細については、4D *ランゲージリファレンス*マニュアルのWebサービス (クライアント)およびXML DOMテーマを参照してください。

# データベースのバックアップと復元


はじめに

バックアップの開始

バックアップスケジュール設定

バックアップファイル設定

バックアップと復旧設定

自動バックアップ

ログファイルの管理

データベースの復元

バックアップジャーナル

## はじめに

4Dには、データベースのフルバックアップと復元用のモジュールが組み込まれています。

このモジュールにより、現在使用中のデータベースを終了しなくても、そのバックアップを作成することができます。各バックアップには、ストラクチャーファイル、データファイル (およびすべてのセグメント)、任意の追加ファイルやフォルダーが含まれます。まず初めに、アプリケーションの環境設定でこれらのパラメーターを設定します。

バックアップの開始は、手動で実行するか、またはユーザーが操作しなくても定期的に自動実行することができます。特定のランゲージコマンドや特定のデータベースメソッドを使用すると、独自のインタフェースにバックアップ機能を統合することができます。

損傷したデータベースが開かれると、自動的にデータベースが復元されます。

また、この統合バックアップモジュールではログファイルを利用することができます。このファイルにより、データに上で実行された全操作の記録が保管され、2回のバックアップ間の安全性が完全に保証されます。使用中のデータベースに問題が発生した場合は、次回そのデータベースを開いた時に、データファイルから失われた全操作が自動的に再統合されます。ログファイルの内容はいつでも確認することができます。

統合されたバックアップモジュールにより、次の操作を行うことができます:

- 随時、データベースファイルのフルバックアップを開始する (ストラクチャーファイル、ログファイル、同封ファイルなど)。
- 定期的な自動バックアップを導入する (時間単位、日単位、週単位、月単位)。
- 上級バックアップパラメーターを設定する (世代数、ファイル圧縮、復元後の再起動オプションなど)。
- 障害の発生時に、データベースと同封ファイルを自動的に復元する。
- ログファイルに保存された不足分の処理を復元後のデータベースへ自動的に統合する。
- データベースのデータに関するロールバック操作を実行する。

**注:** バックアップを目的にデータを複製・同期することで、常に複数の完全一致するデータベースを管理したい場合には、他のソリューションを採用することもできます。これらのソリューションは次のメカニズムや技術に基づいています:

- 4D Serverを使い、(統合されたバックアップモジュールメカニズムを使用して) 論理ミラーを設定する: *4D Server - リファレンス*マニュアルの論理ミラーの設定を参照。
- SQL による同期: *4D - SQLリファレンス*マニュアルのSQLを使用した複製を参照。
- HTTP による同期: 4D *ランゲージリファレンス*マニュアルのURL 4DSYNC/を参照。

## バックアップの開始

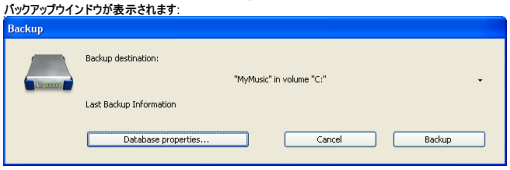
4Dでは、次の3つの方法でバックアップを開始することができます:

- **手動による方法**: 4Dの**ファイル**メニューから**バックアップ...**コマンドを使用します。または、Maintenance & Security Center (MSC) の**バックアップ**ボタンをクリックします。
- **自動的に行う方法**: アプリケーションの環境設定で設定できるスケジューラーを使用します。
- **プログラムによる方法**: BACKUPコマンドを使用します。

どの方法を採用するかは、バックアップ方針によります。

**4D Server:** リモートマシンからBACKUPを呼び出すメソッドを使用して、手動でバックアップを開始することができます。いかなる場合でも、このコマンドはサーバー上で実行されます。

手動バックアップを実行するには、次の手順に従ってください:

1. 4Dの**ファイル**メニューから**バックアップ...**コマンドを選択します。
   バックアップウインドウが表示されます:
   
   
   "バックアップファイルの保存先"エリアの隣のポップアップメニューを使用して、バックアップファイルの保存場所を確認することができます。この場所は環境設定のバックアップ/設定ページにて、設定されています。
   または
   4Dの**ヘルプ**メニューから **Maintenance & Security Center** を選択し、バックアップページを表示させます。
   MSCについての詳細は、Maintenance & Security Centerを参照ください。
   **データベースプロパティ**ボタンで環境設定のバックアップ/設定ページを表示させることができます。
2. **バックアップ**をクリックし、現在のパラメーターを用いてバックアップを開始します。

定期的に自動バックアップを実行するには、次の手順に従ってください:

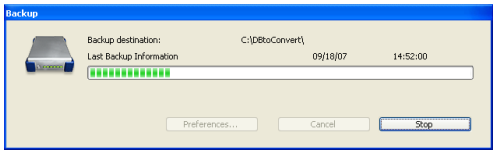
1. 環境設定のバックアップ/スケジューラーページにおいて、バックアップの周期を設定します。
   ユーザーが一切操作しなくても、このページで指定された時間にバックアップが自動実行されます。このダイアログボックスの使用方法に関する詳細は、バックアップスケジュール設定を参照してください。

4Dランゲージを使用して定期的バックアップを実行するには、次の手順に従ってください:

1. メソッドでBACKUPコマンドを実行します。
   現在のパラメーターを用いてバックアップが開始します。
   バックアッププロセスを処理するため、On Backup StartupデータベースメソッドおよびをOn Backup Shutdownデータベースメソッド使用することができます。 (詳細は、4D ランゲージリファレンスマニュアルを参照してください)

### バックアップの実行

バックアップが開始すると、4Dはバックアップの進捗状況を知らせるダイアログボックスを表示します:




MSC を使用している場合、進捗状況は MSC の"バックアップ"ページに表示されます。

**中止** ボタンをクリックすると、いつでもバックアップを中断することができます (バックアップ中に問題が発生した場合参照)。

前回のバックアップの結果 (成功または不成功) は、MSC のバックアップページの"前回のバックアップの情報"エリア、または4D Serverのメンテナンスページで確認できます。また、データベースのバックアップジャーナルにも記録されます。

#### バックアップ中にデータベースへアクセスする

バックアップ実行中のデータベースへのアクセスは、4Dによって制限されます。4Dはバックアップに含まれたファイルタイプに関連するプロセルはすべてロックします: ストラクチャーファイルだけがバックアップされている場合、ストラクチャーにはアクセスできませんが、データにはアクセス可能です。

反対に、データだけがバックアップされているのであれば、ストラクチャーへのアクセスは許可されます。この場合に、データベースへのアクセスが可能かどうかを次に示します:

- シングルユーザー版の4Dの場合、データベースの読み込み、書き込みともにロックされ、すべてのプロセスが停止します。実行できるアクションはありません。
- 4D Server の場合、データベースへの書き込みだけがロックされます。クライアントマシンはデータを照会することができます。クライアントマシンからサーバーへ追加、削除、または変更のリクエストが送信されると、ウインドウが表示され、バックアップの終了まで待機するよう要求されます。
  データベースが保存されるとウインドウが閉じられ、要求したアクションが実行されます。バックアップの終了まで待機せずに、処理中のリクエストをキャンセルするには、**処理をキャンセル**ボタンをクリックしてください。
  ただし、バックアップ前に開始したメソッドから要求されたアクションが実行待機中である場合、このアクションをキャンセルすべきではありません。この場合、実行すべき残りの処理だけがキャンセルされてしまうためです。しかも、メソッドの一部は実行済みなので、データベースにおいて論理上の不整合が生じる可能性があります。
  **注:** 実行待機中のアクションが、メソッドから要求されたものである場合に、ユーザーが処理をキャンセルボタンをクリックすると、4D Server はエラー-9976(データベースのバックアップが進行中なので、このコマンドは実行されません) を返します。

#### バックアップ中に問題が発生した場合

バックアップが正常に実行されない場合もあります。バックアップが不成功に終わる原因としては、ユーザーによる中断、同封ファイルが見つからない場合、保存先のディスクのトラブル、不完全なトランザクションなど、いくつか考えられます。4Dは原因に応じて問題に対処します。

すべての場合において、前回のバックアップのステータス (成功または不成功) は、MSC のバックアップページの"前回のバックアップの情報"エリア、4D Serverのメンテナンスページ、およびデータベースのバックアップジャーナルに表示されます。

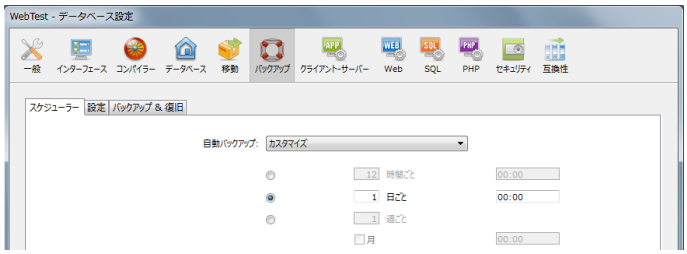
- **ユーザーによる中断**: 進捗ダイアログボックスの**中止**ボタンをクリックすると、いつでもバックアップを中断することができます。この場合、各項目のコピーが中止されてエラー1406が生成されます。このエラーはOn Backup Shutdownデータベースメソッドで遮ることができます。
- **同封ファイルが見つからない**: 同封ファイルが見つからない場合、4Dはバックアップを部分的に実行し (データベースファイルおよび使用可能な同封ファイルのバックアップ)、エラーを返します。
- **バックアップ不可能** (ディスクフル、書き込み保護、ディスクが見つからない、ディスク障害、不完全なトランザクション、定期的な自動バックアップ時にデータベースが起動されていない、など):
  - 初回のエラーの場合、4Dはもう一度バックアップの実行を試みます。この2回のバックアップ間の待機時間は、環境設定のバックアップ/バックアップ＆復旧ページで指定します。
  - 再試行にも失敗した場合、システムの警告ダイアログボックスが表示されてエラーが生成されます。このエラーはOn Backup Shutdown データベースメソッドで遮ることができます。

## バックアップスケジュール設定

4Dや4D Serverで開いたデータベースのバックアップを自動化することができます (4Dクライアントが接続されている必要はありません)。これはバックアップ周期 (時間、日、週、月単位等) を設定することにより行います。設定に基づき、4Dは自動でバックアップを実行します。

バックアップが実行されるべきときにアプリケーションが起動されていなかった場合、次に4Dが起動されたとき、4Dはバックアップが失敗したものと認識し、データベース設定の再試行設定を適用します (バックアップ中に問題が発生した場合参照)。

バックアップのスケジュール設定はデータベース設定のバックアップ/スケジューラーページで行います:



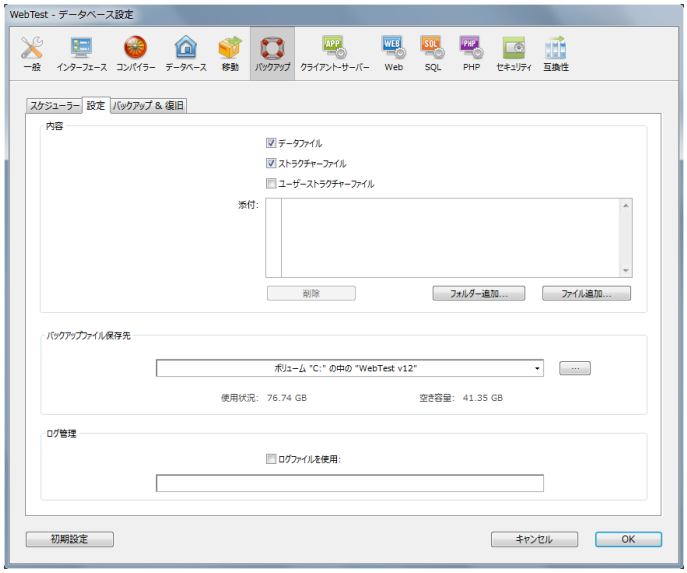

このページにあるオプションを使用して、データベースの自動バックアップのスケジュールを設定できます。標準のクイック設定、または完全なカスタマイズを選択できます。

**自動バックアップ**メニューでの選択に基づき、さまざまなオプションが表示されます:

- **しない**: スケジュールに基づくバックアップは無効となります。
- **毎時**: 次の時間以降、毎時間ごとに自動バックアップを行います。
- **毎日**: 日に一回自動バックアップを行います。バックアップを何時に行うかを設定します。
- **毎週**: 週に一回自動バックアップを行います。バックアップを開始する曜日と時刻を入力するエリアが表示されます。
- **毎月**: 月に一回自動バックアップを行います。バックアップを開始する日付と時刻を入力するエリアが表示されます。
- **個別**: 自動バックアップを詳細にスケジュールする場合に使用します。このオプションを選択すると、複数の入力エリアが表示されます:
  - **時間ごと**: 時間単位でバックアップの間隔をスケジュールできます。1から24までの値を設定できます。
  - **日ごと**: 日単位でバックアップの間隔をスケジュールできます。例えば1を設定すると毎日バックアップが行われます。このオプションを選択した場合、バックアップが開始される時刻を設定しなければなりません。
  - **週ごと**: 週単位でバックアップの間隔をスケジュールできます。例えば1を設定すると毎週バックアップが行われます。このオプションを選択した場合、バックアップを開始する曜日と時刻を設定しなければなりません。複数の曜日を選択できます。例えば毎週水曜日と金曜日にバックアップをするようプログラムできます。
  - **月ごと**: 月単位でバックアップの間隔をスケジュールできます。例えば1を設定すると毎月バックアップが行われます。このオプションを選択した場合、バックアップを開始する日付と時刻を設定しなければなりません。

## バックアップファイル設定

データベース設定のバックアップ/設定ページではバックアップやログファイルの有効化/無効化、および保存先を設定できます。これらのパラメーターは4Dアプリケーションで開かれる各データベースごとに設定されます。




**4D Serverに関する注記:** これらのパラメーターは4D Serverマシン上でのみ設定できます。

### 内容

このエリアでは次回のバックアップ時にコピー対象とするファイルやフォルダーを指定します。

- **データファイル**: データベースデータファイル。このオプションが選択されている場合、(存在すれば) データベースのカレントログファイルがデータとともにバックアップされます。
- **ストラクチャーファイル**: データベースストラクチャーファイル。コンパイルされ4D Volume Desktopが結合されたデータベースの場合、このオプションを選択するとWindowsでは.exeファイルが、Mac OSではパッケージがバックアップされます。
- **ユーザーストラクチャーファイル** (任意): カスタマイズされたユーザーフォームが格納されたユーザーストラクチャーファイル (作成されている場合)。詳細はユーザーフォームを参照してください。
- **添付**: このエリアではデータベースと同時にバックアップの対象とするファイルやフォルダーを指定します。ここではどのようなタイプのファイル (ドキュメントやプラグイン、テンプレート、ラベル、レポート、ピクチャーなど) でも指定できます。
  ファイルまたはその内容をバックアップするフォルダーを個々に設定できます。添付エリアには設定されたファイルのパスが表示されます。
  - **削除**: 選択したファイルを添付エリアから取り除きます。
  - **フォルダー追加...**: バックアップに追加するフォルダーを選択するダイアログボックスを表示します。復元の場合、フォルダーがその内容物とともに復元されます。データベースファイルを含むフォルダーを除き、すべてのフォルダーやマシンに接続されたボリュームを選択できます。
  - **ファイル追加...**: バックアップに追加するファイルを選択するダイアログボックスを表示します。

4Dデータベースファイルについては4Dファイルの説明を参照してください。

### バックアップファイル保存先

このエリアではバックアップファイルの格納場所を確認したり、変更したりできます。

エリアをクリックすると、ファイルの場所がポップアップで表示されます。

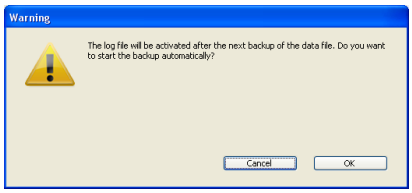
バックアップファイルの格納場所を変更するには、[...]ボタンをクリックします。選択ダイアログが表示され、バックアップファイルを配置するフォルダーやディスクを選択できます。"使用状況"と"空き容量"エリアは、選択したフォルダーが存在するディスクの状態を自動で表示します。

### ログ管理

**ログファイルを使用**オプションが選択されていると、データベースはログファイルを使用します。ログファイルの場所はオプションの下に表示されます。このオプションが選択されている場合、ログファイルなしでデータベースを開くことはできません。

4Dで作成されたデータベースはデフォルトでログファイルを使用します。このファイルの作成は自動バックアップの計画に含まれています (自動バックアップ参照)。ログファイルには*DataFileName.journal*のように名前が付けられ、データベースストラクチャーと同階層に置かれます。

新しいログファイルを有効にするためには、その前にデータベースのデータをバックアップしなければなりません。このオプションを選択してデータベース設定ウィンドウを閉じると、バックアップが必要である旨の警告ダイアログが表示されます:

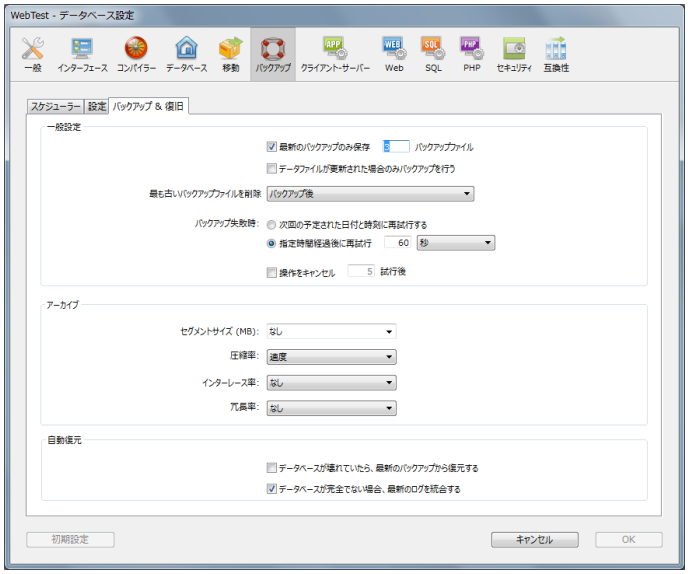



**OK**をクリックすると即座にバックアップが開始され、ログファイルが有効になります。**キャンセル**をクリックすると設定は保存されますが、ログファイルの作成は延期されます。この場合ログファイルは次回のバックアップ後に作成されます。

## バックアップと復旧設定

バックアップ設定の変更は必要に応じて行います。デフォルトで標準的なバックアップ動作を行うようになっています。

バックアップのオプション設定はデータベース設定の[バックアップ/バックアップ＆復旧ページ](#)で行います:




### 一般設定

このエリアでは実際にバックアップが行われる際に実行される様々なメカニズムの設定を行います。

- **最新のバックアップのみ保存 X バックアップファイル**: このパラメーターを有効にすると、指定された数の最新バックアップファイルだけが保持され、古いバックアップファイルは削除されます。これによりディスクドライブの容量が不足することを避けることができます。
  この機能は以下のように動作します: バックアップ処理が完了したら、アーカイブが作成されたのと同じ場所、同じ名前のもっとも古いアーカイブを削除します。削除のタイミングを変更することもできます。
  例えば3世代のファイルを保持するよう設定している場合、最初の3回のバックアップでMyBase-0001、MyBase-0002、MyBase-0003が作成され、四回目のバックアップでMyBase-0004が作成されたのちにMyBase-0001が削除されます。デフォルトでこの設定は有効になっていて、4Dは3世代のバックアップを保持します。
  このメカニズムを無効にするには、チェックボックスの選択を外します。
  **注:** このパラメーターはデータベースバックアップセットとログファイルバックアップセット両方に関連します。
- **データファイルが更新された場合のみバックアップを行う**: このオプションが選択された場合、前回のバックアップ以降にデータが追加、変更、削除された場合のみ4Dは定期的なバックアップを開始します。そうでない場合定期的なバックアップはキャンセルされ、次回のスケジュールまで遅延されます。エラーは生成されませんが、バックアップジャーナルにはバックアップが遅延された旨記録されます。
  このオプションを使用すれば、原則的に参照目的で使用されているデータベースのバックアップに消費されるマシン時間を節約できます。ストラクチャーや添付ファイルに対して行われても、データファイルの更新としては扱われない旨注意してください。
  **注:** このパラメーターはデータベースおよびログファイル両方のバックアップに影響します。
- **最も古いバックアップファイルを削除**: このオプションは"最新のバックアップのみ保存 X バックアップファイル"が有効になっている場合のみ考慮されます。このオプションを使用して、最も古いバックアップファイルを削除するタイミングを設定します。選択肢はバックアップの開始前、あるいは完了後です。このオプションが有効に働くためには、バックアップファイルが名称変更されたり、移動されたりしていてはなりません。
- **バックアップ失敗時**: このオプションを使用してバックアップ失敗時の処理を設定できます。
  **注:** 定期的なバックアップが実行される予定時刻にデータベースが起動されていなかった場合、4Dはバックアップが失敗したものとして扱います。
  - **次回の予定された日付と時刻に再試行する**: このオプションは定期的な自動バックアップを設定されている場合にのみ意味があります。
  - **指定時間経過後に再試行**: このオプションが選択されていると、設定された待ち時間経過後にバックアップを再試行します。このメカニズムを使用すると、バックアップをブロックするような特定の状況に対応することが可能となります。秒、分、あるいは時間単位で待ち時間を設定できます。次のバックアップ試みにも失敗するとエラーが生成され、ステータスエリアに失敗状況が表示され、バックアップジャーナルに記録されます。
  - **操作をキャンセル X 試行後**: このパラメーターを使用して、バックアップ試行の失敗最大数を設定できます。
    この最大数に達してもバックアップが正しく実行できなかった場合、バックアップはキャンセルされエラー1401 ("バックアップ試行の最大数に達しました。自動バックアップは無効になります") が生成されます。この場合、データベースを再起動するか、手動バックアップが成功するまで自動バックアップは行われません。
    このパラメーターは、人による介入が必要となるような問題があり、バックアップ試行が自動的に繰り返されることにより全体的なパフォーマンスに影響するようなケースで使用できます。デフォルトでこのオプションは選択されていません。

### アーカイブ

このエリアではアーカイブ生成オプションを設定します。これらのオプションはメインのバックアップファイルとログバックアップファイルに適用されます。

- **セグメントサイズ (MB)**
  4Dではアーカイブをセグメントに分割できます。この振る舞いにより例えばバックアップファイルを複数の異なるディスク (DVDやZIP等) に格納できます。復元時、4Dは自動でセグメントを統合します。各セグメントにはMyDatabase[xxxx-yyyy].4BKといった名称がつけられます (xxxxはバックアップ番号、yyyyはセグメント番号)。例えばMyDatabaseデータベースのバックアップが3つのセグメントに分割されると、以下のような名前になります:
  MyDatabase[0006-0001].4BK、MyDatabase[0006-0002].4BK、MyDatabase[0006-0003].4BK
  **セグメントサイズ**はコンボボックスであり、各セグメントのサイズをMB単位で設定できます。メニューから定義済み値を選択するか、0~2048の値を入力できます。0を指定するとセグメント化はされません (**なし**を指定したのと同じ)。
- **圧縮率**
  デフォルトで4Dはバックアップファイルを圧縮し、ディスクスペースを節約します。しかし大量のデータがある場合、ファイルの圧縮処理はバックアップにかかる時間を長くします。
  **圧縮率**オプションを使用してファイルの圧縮モードを調整できます:
  - **なし**: ファイルの圧縮は行われません。バックアップは早く行われますが、ファイルサイズは大きくなります。
  - **速度** (デフォルト): このオプションはバックアップの速度とアーカイブサイズのバランスが考慮されたものです。
  - **圧縮率**: アーカイブに最大の圧縮率が適用されます。アーカイブファイルはディスク上で最小のサイズとなりますが、バックアップの速度は低下します。
- **インターレース率と冗長率**
  4Dは最適化 (インターレース) とセキュリティ (冗長) メカニズムに基づく特定のアルゴリズムを使用してアーカイブを生成します。これらのメカニズムを必要に応じて設定できます。これらのオプションのメニューには**低**、**中**、**高**、および**なし** (デフォルト) の選択肢があります。
  - **インターレース率**: インターレースとはデータを連続しない領域に書き込むことにより、セクター損傷の際のリスクを低減させるものです。率を上げることでリスクがより低減されますが、データの処理により多くのメモリが必要となります。
  - **冗長率**: 冗長は同じ情報を複数回繰り返すことで、ファイル中のデータを保護するものです。冗長率を高くするとよりファイルが保護されます。しかし書き込みは遅くなり、ファイルサイズも増大します。

### 自動復元

ここでは損傷したデータベースを開く際の自動メカニズムを設定します。

- **データベースが壊れていたら、最新のバックアップから復元する**: このオプションが選択されていると、異常が検知された場合、4Dは起動時に自動でデータベースの有効な最新のバックアップからのデータの復旧を開始します。ユーザーによる介入は必要ありませんが、処理はバックアップジャーナルに記録されます。
  **注:** 自動復元が行われた場合、データファイルのみが復元されます。添付ファイルやストラクチャーファイルを取得したい場合、手動の復元を行う必要があります。
- **データベースが完全でない場合、最新のログを統合する**: このオプションがチェックされると、プログラムはデータベースを開く際や復旧時に、自動でログファイルを統合します。
  - データベースを開く際に、データファイルに保存されていない処理がログファイル中に見つかった場合、4Dは自動でカレントログファイルを統合します。このようなケースはディスクに書き込まれていないデータがまだキャッシュ中に存在する状態で、電力の切断が起きた場合に発生します。
  - データベースの復元時、カレントログファイルや、バックアップファイルと同じ番号を持つログバックアップファイルが同じフォルダーに存在する場合、4Dはその内容を検証します。そしてデータファイルに書き込まれていない処理が見つかれば、自動で統合が行われます。

ユーザーにダイアログボックスが提示されることはありません。処理は完全に自動です。処理はバックアップジャーナルに記録されます。

## 自動バックアップ

4Dはデータベースの作成時に、デフォルトのバックアップ方法を提供します。このバックアップ方法により一連のバックアップパラメーターが設定され、最低限のデータベースセキュリティーが保証されます。

これらの設定は、データベースの作成後に変更することができます。

データベースを作成すると、このバックアップ方法がデフォルトで有効に設定されます。無効にするには、環境設定の一般ページの自動バックアップをすべての新規データベースで有効にするオプションの選択を解除します。

デフォルトの自動バックアップ方法には、次の設定が含まれています:

- 毎日曜00:00:00時の自動バックアップ
- データベースストラクチャーファイルと同じ階層にあるすべてのデータベースファイルをバックアップする (データ、ログ、ストラクチャー、ユーザーストラクチャー)。
- ログファイルを作成し、使用する (ファイル名は、データベース名.journal)。
- デフォルトのバックアップ設定 (最新のバックアップを3つ保管、圧縮率"中"など)
- すべての自動復元オプション

これらのパラメーターはデータベース作成後にカスタマイズすることが出来ます。

# ログファイルの管理

継続的に使用されるデータベースでは、変更の記録およびレコードの追加や削除が常に行われています。定期的にデータをバックアップすることは重要ですが、バックアップだけでは (予期しない障害の場合に)、前回のバックアップ以降に入力されたデータを回復することができません。この必要性に対応するため、4Dは特定のツールであるログファイルを提供しています。このファイルを使用すると、データベースデータのセキュリティーが常に保証されます。

また、4Dは常にメモリー上のデータキャッシュを使用して作業を行います。データベースのデータへの変更はすべて、ハードディスクへ書き込む前に、キャッシュへ一時的に保存されます。これにより、アプリケーションの処理速度が向上します。実際、メモリーへのアクセスは、ハードディスクへのアクセスよりも高速に行われます。キャッシュに保存したデータをディスクへ書き込む前に障害が発生した場合は、カレントログファイルを組み込んでデータベースを完全に復旧しなくてはなりません。

さらに、4Dには、カレントログファイルの内容を解析する機能が組み込まれており、これによってデータベースのデータ上で実行されたすべての処理をさかのぼることができます。これらの機能は MSC にて提供されています (アクティビティ解析ページおよびロールバックページ参照)

**はじめに**

4Dが生成するログファイルには、データベース上で行われたすべての操作が順次記録されています。デフォルトでは、全てのテーブルはデータログが取られています(つまりログファイルに含まれています)。しかし [#title id="651" anchor="1282124"/] のプロパティを使用することによって特定のテーブルをログに含めないようにすることもできます。

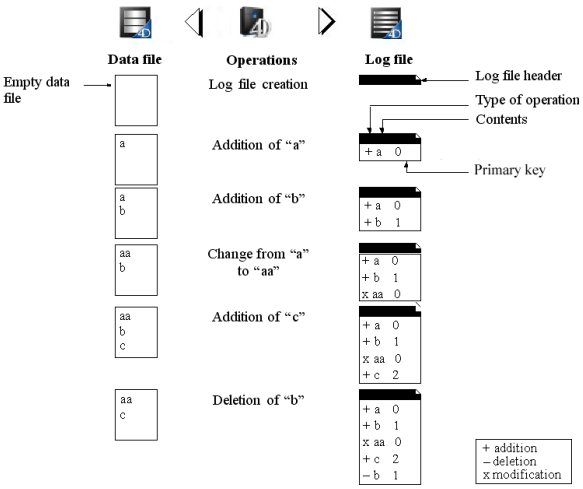
したがって、ユーザーが実行した各操作により、2 つのアクションが同時に行われます。1つは、データベースに対するアクション (命令を通常どおりに実行)、もう1つはログファイルに対するアクション (処理の説明を記録) です。ログファイルは個別に作成され、ユーザーの作業を妨げたり作業速度を低下させることはありません。1つのデータベースでは、一度に1つのログファイ ルだけを扱うことができます。ログファイルには、次の操作が記録されます:

- データファイルの開閉
- プロセス (コンテキスト) の開閉
- レコードまたはBLOBの追加
- レコードの変更
- レコードの削除
- トランザクションの作成や終了

これらのアクションについての詳細は、アクティビティ解析ページを参照ください。

4D はログファイルを取り扱います。ログファイルは、データファイルに影響を与えるすべての操作を区別なく盛り込み、ユーザーが行った操作や4Dメソッドによる処理、4D SQL エンジン、4Dプラグイン (4D Write、4D View など)、Web ブラウザによる処理など、あらゆる操作を記録します。

ログファイルの機能をまとめた図を次に示します:




カレントログファイルはカレントデータファイルと一緒に自動保存されます。このメカニズムには、2つの際立った利点があります。

- ロ グファイルが保存されるディスクの容量が一杯にならないようにします。バックアップを実行しない場合、ログファイルは使用するにつれて徐々に大きくなり、いずれはディスクの空き容量をすべて使い果たしてしまいます。データファイルをバックアップするたびに、4Dや4D Server はカレントログファイルをクローズし、その直後に空ファイルを新たに開くため、ディスクフルになる危険を避けることができます。この後、古いログファイル はアーカイブに保存され、バックアップのセット (世代) を管理するメカニズムに従って最終的には破棄されます。
- 後からデータベース の解析や修復を行えるように、各バックアップに対応するログファイルを保管します。ログファイルの統合は、それが対応するデータベースからのみ実行できま す。ログファイルをバックアップに正しく統合するために、バックアップとアーカイブ化されたログファイルとを同時に保持することが重要です。

**ログファイルの作成**

デフォルトでは、4Dで作成されたあらゆるデータベースでログファイルが使用されます。このファイルの作成は、自動バックアップオプションに基づいています(自動バックアップ参照)。ログファイルには**データファイル名.journal**という名前が付けられ、データベースストラクチャーを含むフォルダー内に置かれます。

データベースでログファイルが使用されているかどうかは、いつでも調べることができます。これを行うには、環境設定のバックアップ/設定ページで**ログファイルを使用**オ プションが選択されているか確認します ([#title id="1092" anchor="462271"/]参照)。このオプションの選択を解除した場合、またはログファイルなしでデータベースを使用しており、ログファイルを 用いたバックアップ方法を導入したい場合は、ログファイルを作成する必要があります。

ログファイルを作成するには、次の手順に従ってください:

1. 環境設定のバックアップ/設定ページで、**ログファイルを使用**オプションを選択します。ログファイルにはデフォルトで**データファイル名.journal**という名前が付けられます。
2. デフォルトの名前を使用するか、またはその名前を変更し、次にファイルの保管場所を選択します。
   2つ以上のハードドライブが存在する場合は、データベースが保管されているディスク以外の場所にログファイルを保存することをお勧めします。これにより、データベースが保管されているハードドライブが破損した場合でも、ログファイルを呼び出すことができます。
3. **保存**をクリックします。
   開いたログファイルのアクセスパスと名前が環境設定ダイアログボックスの"ログファイルを使用"エリアに表示されます。このエリアをクリックすると、ポップアップメニューが表示され、ディスク上のフォルダーを確認することが出来ます。
4. 環境設定ダイアログボックスを確定します。

**注:** 4D v14 以降、ログファイルを作成・使用するための新しい内部メカニズムが採用されたことにより、このファイルはより抵抗力を持つようになりました。ログファイルを使用するためには、データログを取る全てのテーブルがプライマリーキーを持っている必要があります。そうでない(変換されたデータベース等の)場合には、データログを取る全てのテーブルにおいてプライマリーキーが正常に設定されていない限りログファイルを起動することはできない旨の警告ダイアログボックスが表示されます。この規則に適合するために プライマリーキー管理 を使用することができます。アシスタントは全てのテーブルの状態をチェックし、プライマリーキーの設定を手助けします。

ログファイルの作成を可能にするには、データベースが次の条件のいずれかを満たしていなくてはなりません(ただしプライマリーキーに適合していない場合は除きます。詳細は前項を参照して下さい)。

- データファイルが空である。
- データベースのバックアップを実行した直後であり、データへの変更がまだ行われていない。

いずれの条件も満たしていない場合は、プログラムはバックアップを実行する必要がある旨を知らせる警告ダイアログボックスを表示します。**OK**をクリックすると、バックアップが開始され、その後にログファイルが作成されます。**Cancel**を クリックした場合には、ログファイル作成の要求は保存され、次回データベースをバックアップする時までログファイルの作成は延期されます。このような安全 対策が不可欠な理由は、障害の発生後にデータベースを復元するために、ログファイルへ記録された処理を統合するデータベースのコピーが必要となるからで す。

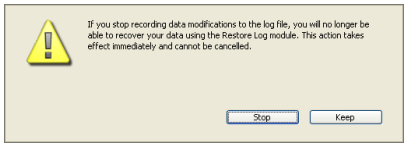
これ以外に何も行わなくても、データ上で実行されたすべての処理がこのファイルに記録され、その後データベースをオープンした時にこのファイルが使用されます。

新規データファイルを作成したら、別のログファイルを作成しなくてはなりません。また、ログファイルに関連付けられていない (または、ログファイルが見つからない) 別のデータファイルをオープンした場合、他のログファイルを設定するか、作成しなくてはなりません。

**ログファイルを中止する**

カレントログファイルへの操作記録を中止したい場合は、アプリケーションの環境設定のバックアップ/設定ページの**ログファイルを使用**オプションを選択解除します。

すると、4Dは警告メッセージを表示して、この動作によりログファイルによるセキュリティが利用できなくなることを知らせます:




**停止** をクリックすると、カレントログファイルが即座にクローズされます (この後に環境設定ダイアログボックスを確定する必要はありません)。

カレントログファイルが大きすぎるため、それをクローズしたい場合は、データファイルのバックアップを実行してください。これにより、ログファイルのバックアップが作成されます。

**4DServer:**

- New log fileコマンドはカレントログファイルを自動的に閉じて、新しいログファイルを開始します。
- 実行中何らかの理由でログファイルが利用不能になった場合、エラー1274が生成され、4D Serverは一切のデータ書き込みを許可しなくなります。再びログファイルが利用可能になったらフルバックアップを実行しなければなりません。

# データベースの復元

問題が発生したときは、一連のデータベースファイル全体を復元することができます。主に2つのカテゴリの問題が発生する可能性があります:

- データベース使用中に予期せず終了した
  この問題は電力の切断、システムのエラー等により発生する可能性があります。この場合、問題が発生した瞬間のデータベースキャッシュの状態により、データベースの復旧には異なる手順が必要となります:
  - キャッシュが空の場合、データベースを問題なく開くことができます。データベースに対して行われた変更はデータファイルに記録されています。この場合には特別な手順は必要ありません。
  - キャッシュに未保存の処理が含まれている場合、データファイルは損傷していませんが、カレントのログファイルを統合する必要があります。
  - キャッシュの内容をデータファイルに書き込み中だった場合、データファイルはおそらく損傷しています。最新のバックアップから復元を行い、カレントのログファイルを統合する必要があります。

- データファイルを失った
  この問題はデータベースが配置されたディスク上のセクターが読み書き不能になった、あるいはウィルス、操作ミス等により発生します。最新のバックアップから復元を行い、カレントのログファイルを統合しなければなりません。問題発生後にデータベースが損傷しているかどうかを見分けるには、4Dを使用してデータベースを起動します。4Dは自己検証を行い、必要な復元処理手順を示します。自動モードの場合、この処理はユーザーのアクションなしで直接行われます (後述参照)。定期的なバックアップが行われていれば、4Dの復元ツールを使用して (ほとんどの場合) 問題が発生する直前の状態までデータベースを復旧することができます。

**注:**

- 問題発生後に、自動で4Dのデータベース復旧処理を起動することができます。このメカニズムはデータベース設定のバックアップ/バックアップ＆復旧ページで利用できる2つのオプションを使用して管理します。詳細は自動復元を参照してください。
- 問題が、データに対して行われた不適切な処理の結果引き起こされた場合 (例えば誤ってレコードを削除した等)、ログファイルの"ロールバック"機能を使用してデータベースを復旧できます。この機能はMSCのロールバックページから利用できます。

### 手動でバックアップから復元する (標準ダイアログ)

バックアップモジュールを使用して生成されたアーカイブの内容を、手動で復元できます。手動による復元は例えば、アーカイブ全体 (ストラクチャーファイルや添付されたファイル) を再生成したい場合や、アーカイブの内容を見たい場合などに必要となります。
手動復元の際に、カレントログファイルを統合することもできます。

バックアップの手動復元は標準のファイルを開くダイアログボックスあるいはMaintenance and Security Center (MSC) の"復元"ページから行うことができます。

- 標準のダイアログボックスを使用した復元は、どのアーカイブに対しても行うことができます。この機能については以下で説明します。
- MSCを使用した復元では詳細なオプション設定を行ったり、アーカイブの内容をプレビューしたりすることができます。他方、開かれたデータベースに関連付けられたアーカイブのみを復元できます。この機能は手動でバックアップから復元する (MSC)で説明しています。

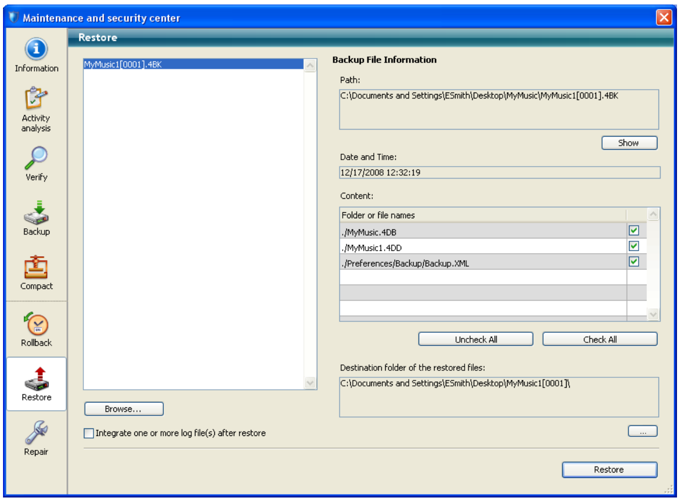
標準ダイアログボックスを使用してデータベースを手動復元するには:

1. 4Dアプリケーションを開始し、**ファイル**メニューから**復元...**を選択します。
   データベースが開かれている必要はありません。
   または
   4DメソッドからRESTOREコマンドを実行します。
   標準のファイルを開くダイアログボックスが表示されます。復元するバックアップファイル (.4bk) またはバックアップログファイル (.4bl) を指定します。
2. 復元するファイルを選択し、**開く**をクリックします。
   復元したファイルを配置する場所を指定するために、以下のダイアログボックスが表示されます:

   
   デフォルトで4Dはアーカイブと同階層にアーカイブ名と同じ名前 (拡張子なし) のフォルダーを作成し、ファイルを復元します。**[...]**ボタンをクリックして異なる場所を指定することもできます。
3. **復元** ボタンをクリックします。
   4Dは指定されたすべてのバックアップファイルを展開します。
   カレントログファイルまたはバックアップファイルと同じ番号のバックアップログファイルが同じ場所にあれば、4Dはその内容を検証します。ログファイルに、データファイル中に無い処理が含まれていれば、その処理を統合するかどうか4Dが尋ねてきます。"データベースが完全でない場合、最新のログを統合する"が選択されている場合、統合処理は自動で行われます (自動復元参照)。
4. (任意) **OK**をクリックして、復元したデータベースにログファイルを統合します。
   復元と統合が正しく実行されると、4Dは処理が成功したことを通知するダイアログを表示します。
5. **OK**をクリックします。
   保存先フォルダーが表示されます。バックアップ時のファイルの位置にかかわらず、4Dはすべてのバックアップファイルをこのフォルダーに配置します。

### 手動でバックアップから復元する (MSC)

Maintenance and Security Center (MSC) の**復元** ページから、カレントデータベースのアーカイブを手動で復元できます。このページではデータベースの復元を制御するためのオプションをいくつか提供します:




ウィンドウの左側にはデータベースの既存のバックアップが表示されます。**選択...**ボタンをクリックして他の場所にあるアーカイブファイルを選択することもできます。選択したアーカイブはリストに追加されます。

このリストからバックアップファイルを選択すると、ウィンドウの右側にはこのバックアップについての説明が表示されます:

- **パス**: 選択されたバックアップファイルの完全パス名。**表示** ボタンをクリックするとシステムウィンドウでバックアップファイルが表示されます。
- **日付と時刻**: バックアップの日付と時刻
- **内容**: バックアップファイルの内容。各項目の右側にはチェックボックスがあり、復元を行うかどうか、ファイルごとに選択できます。**すべてを選択する**や**すべての選択を外す**ボタンを使用して復元するファイルの切り替えを行うこともできます。
- **復元されたファイルの保存先フォルダー**: 復元されたファイルが配置されるフォルダー。デフォルトで4Dはデータベースストラクチャーと同階層にアーカイブ名 (拡張子なし) のフォルダーを作成し、そこにファイルを復元します。個の場所を変更するには**[...]**をクリックして復元ファイルの配置場所を指定します。

#### 複数のログファイルを連続して統合する

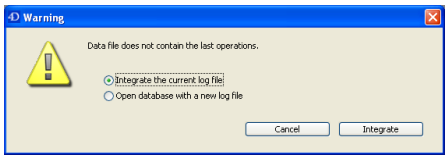
**復元後にひとつ以上のログファイルを統合**オプションを使用して、データベースに複数のログファイルを統合することができます。例えば4つのデータベースバックアップに対応する4つのログファイルアーカイブがある場合、最初のバックアップを復元して、ログアーカイブを一つずつ統合できます。これにより例えば最新のバックアップファイルを失った場合でも、データファイルを復旧することができます。

このオプションが選択されていると、4Dは復元後に標準のファイルを開くダイアログを表示します。ここで統合するログファイルを選択できます。ファイルを開くダイアログはキャンセルされるまで、統合毎に表示されます。

### 手動でログを統合する

MSCの**復元** ページでログファイルの統合を選択していない場合 (複数のログファイルを連続して統合する参照)、データファイルに保存されていない処

理がログファイル中に見つかった場合、4Dはデータベースを開く際に警告ダイアログボックスを表示します。




このメカニズムを機能させるために、4Dはカレントの場所にあるログファイルにアクセスできなければなりません。

カレントログファイルを統合するかしないかを選択することができます。カレントログファイルを統合しないことにより、データ中に作成されたエラーを再生成しないようにすることもできます。

# バックアップジャーナル

### バックアップジャーナル

データベースのバックアップの追跡や検証を容易にするため、バックアップモジュールは実行された各処理の概要を特別なファイルに書き込みます。このファイルは、いわゆる活動記録のようなものです。オンラインマニュアルのように、処理が定期的または手動のいずれで行われていても、すべてのデータベース操作 (バックアップ、復元、ログファイルの統合) がこのファイルに記録されます。これらの処理が実行された日付と時刻もこのジャーナルに記述されます。

バックアップジャーナルには"Backup Journal.txt"という名前が付けられ、データベースの"Logs"フォルダーに配置されます。

バックアップジャーナルは、任意のテキストエディターで開くことができます。解析を容易にするため、ジャーナルの情報はタブで区切られ、情報の各行は改行で終了しています

#### バックアップジャーナルのサイズ管理

バックアップ方法によっては、バックアップジャーナルのサイズがすぐに大きくなってしまうことがあります (例えば、同封ファイルが一緒にバックアップされる場合)。

このサイズを管理するには、2つの方法があります:

- **自動バックアップ:** 4Dはバックアップを実行する前にカレントバックアップジャーナルファイルのサイズを確認します。10MBよりも大きい場合、カレントファイルはアーカイブされ、新しいファイルを作成します。アーカイブされたファイルの名前は"Backup Journal[xxx].txt"に変更されます (xxxには1から999までの数字が入ります)。ファイル番号が999を超えると、ナンバリングは1に戻り、既存ファイルが置換されます。
- **記録する情報量を削減する:** このためには、データベースの Backup.xml ファイルの VerboseMode キーの値を変更します。デフォルトでは、True の値が設定されています。この値を False に変更すると、バックアップジャーナルには主要な情報のみが記録されます (スタート時の日付と時刻、そしてエラーの有無)。バックアップ設定に関する XML キーは *XML キー - バックアップマニュアル* (PDF) にて説明されています。

### Backup.xml

4Dのバックアップ設定はデータベースに関連する XML ファイルとして提供されています。バックアップを開始する際、4Dはこのファイルのデータに基づいてバックアップオプションを指定します (これらのオプションは環境設定ダイアログボックスにあります)。このファイルは、バックアップジャーナルに保存する情報量などの追加オプションを設定するのにも使用できます。

バックアップ設定の XML ファイル名は **Backup.xml** です。このファイルは自動的に次の場所に作成されます:

*データベースフォルダー*/Preferences/Backup/Backup.xml

上の*データベースフォルダー*とは、データベースストラクチャーファイルを格納しているフォルダーのことです。Preferences/Backup というサブフォルダーは必要に応じて自動作成されます。

バックアップ設定に関する XML キーは *XML キー - バックアップマニュアル* (PDF) にて説明されています。

## Maintenance & Security Center


概要

情報ページ

アクティビティ解析ページ

検証ページ

バックアップページ

圧縮ページ

ロールバックページ

復元ページ

修復ページ

## 概要

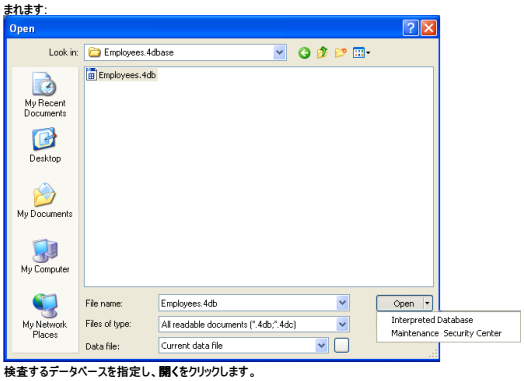
Maintenance & Security Center (MSC) には、データとストラクチャーファイルを検証、保守、バックアップそして圧縮するツールを提供します。MSC は4Dシングルユーザー、4D Server、そして4D Desktopなどすべての4Dアプリケーションから利用できます。

MSC ウィンドウを開く方法は幾つかあります。アクセスの方法により、"メンテナンス" モードまたは" 標準" モード、いずれによってデータベースを開くかも決定されます。メンテナンスモードの場合、4D はデータベースを開かず、その参照だけがMSC に供給されます。標準モードの場合、4D はデータベースを開きます。

### メンテナンスモードでの表示

メンテナンスモードでは、MSC ウィンドウだけが表示されます (4D アプリケーションはデータベースを開きません)。つまり、損傷が激しいため4Dが標準モードで開けないデータベースにもアクセスできるということです。さらに、特定の操作 ( 圧縮、修復など) はデータベースをメンテナンスモードで開くことを要求します (アクセス権参照)。

メインテナンスモードの場合、次の2 つの場所からMSC を開くことができます:

- **標準の開くダイアログボックス**
  標準のデータベースを開くダイアログボックスには**開く**ボタンに関連付けられているメニューに**Maintenance & Security Center**オプションが含まれます:
  
  
  検査するデータベースを指定し、**開く**をクリックします。
- **Help/Maintenance Security Center** メニューまたはツールバーの **MSC** ボタン (データベースが開かれていない状態)

  
  この機能を呼び出すと、標準のファイルを開くダイアログボックスが表示され、検査するデータベースを指定できます。データベースは開かれません。

### 標準モードでの表示

標準モードではデータベースが開いています。このモードでは、特定の保守機能を使用できません。この状況でMSCを開く方法が幾つかあります:

- **Help/Maintenance Security Center**メニューまたはデザインモードでのツールバーの**MSC**ボタンを使用する。
  このコマンドはMSC ウィンドウを開きます。この機能はアプリケーションモードにはありません。
- "MSCを開く"標準アクションを割り当てたメニューコマンドやボタンを使用する (標準アクション参照)。
- OPEN SECURITY CENTERランゲージコマンドを使用する。

### アクセス権

MSC の特定の機能はアプリケーションのタイプ、MSC が開かれたモード、または ( パスワードシステムが有効な場合) ユーザプロファイルにより、無効化されます:

- アプリケーションストラクチャーに影響を及ぼす機能 (確認、修復および圧縮) は、ローカルモードの4Dおよび4D Server アプリケーションからのみアクセスできます。
  リモートモードの4Dおよび4D Desktopアプリケーションではこれらのボタンやタブが隠されます。
- データおよびストラクチャーの内容に関する情報は、データベースが開かれている時(MSC が標準モードで開かれているとき) のみ提供されます。
- データの圧縮、ロールバック、復元および修復の機能は、開いていないデータファイル(MSC はメインテナンスモードで開かれていなければなりません) にのみ使用できます。データベースが標準モードで開かれている時にこれらの機能を試みた場合は、メインテナンスモードでアプリケーション再起動を促すダイアログボックスが表示されます。
- パスワードが有効化されている場合、データの圧縮、ロールバック、復元および修復の機能へはAdministrator およびDesignerだけがアクセスできます。

## 情報ページ

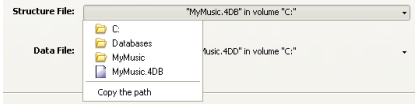
情報ページは4D 環境、システム環境、データベースおよびアプリケーションファイルについての情報を提供します。各ページは、ウィンドウ上部にあるタブコントロールを使って切り替えできます。

### プログラム

このページにはアプリケーションならびにアクティブな4Dフォルダーの名前、バージョンおよび場所を表示します (アクティブ4Dフォルダーについては4Dランゲージリファレンスの[Get 4D folder](#)コマンドを参照)。

ウィンドウの中央部は、データベーストラクチャーならびにデータファイルの名前および場所を表示します。ウィンドウの下部は、4D ライセンスフォルダーの名前、ライセンスのタイプ、およびパスワードが有効化されている時のデータベースユーザの名前 (有効でない場合はDesigner) を表示します。

- **パス名の表示と選択**: **プログラム**ページには、パス名はディスク上の一連のフォルダーで構成されるポップアップメニューで表示されます:




  メニュー項目 ( ディスクまたはフォルダー ) を選択した場合、そのパスが新しいシステムウィンドウで開かれます。
  **パスをコピー**コマンドは、システムのディレクトリ区切り文字を使用して、完全なパス名をクリップボードにテキストとしてコピーします。

- **"Licenses"フォルダー**
  **"Licenses" フォルダー** ボタンをクリックすると、新しいシステムウィンドウにアクティブなLicensesフォルダーの中身を表示します。インストールされた4D環境用のライセンスファイルはすべてこのフォルダーに格納されていなければなりません。ファイルをWebブラウザーで開くと、ライセンスの内容が表示されます。
  "Licenses"フォルダーの場所はバージョンやOSにより異なります。このフォルダーの場所については[Get 4D folder](#)コマンドの説明を参照してください。
  **注**: ヘルプメニュー内ライセンス更新メニューコマンドからアクセスできるダイアログボックスにも同じボタンがあります。
  4Dアプリケーションをアクティベーションしている場合、このフォルダー内に少なくとも1つのライセンスファイルが存在するはずです。複数の製品や拡張ライセンスをアクティベーションすると、その数だけライセンスファイルが作成されます。

### テーブル

データベースの全テーブル、ならびにそれらの特徴(各テーブルの番号、各テーブルのレコード総数、フィールド総数およびインデックス総数) を表示します。

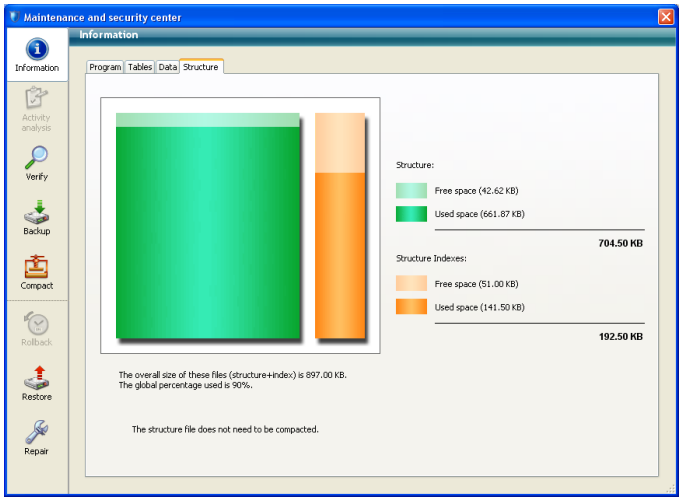
### データとストラクチャー

**データ**および**ストラクチャー**ページにはデータベーストラクチャーファイルとデータファイルの、全体ファイルサイズに対する実使用サイズの情報が表示されます。

**注**:

- これらのページにはメンテナンスモードではアクセスできません。
- **データ**ページに表示される情報には、データファイル外に格納されたデータは反映されません ([データをデータファイル外に保存](#)参照)。
- **ストラクチャー**ページは4Dおよび4D Serverアプリケーションでのみ利用できます。

この情報はグラフ形式で提供されます:



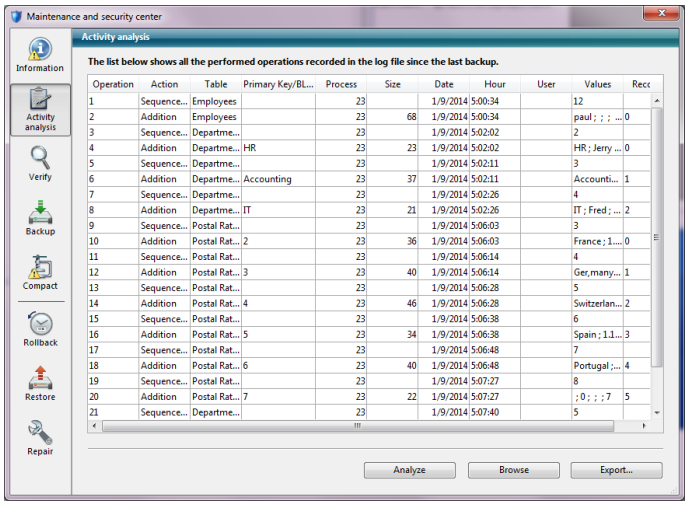

断片化があまりにも進んだファイルはディスク、そしてデータベースのパフォーマンスを低下させます。使用率が低すぎる場合、4Dは警告アイコンを使用してこのことを示します (このアイコンは**情報**ボタンと対応するファイルタイプのタブに表示されます)。そして圧縮が必要であることを警告します。:

警告アイコンは[GET STYLE SHEET INFO](#)ボタンにも表示されます:

## アクティビティ解析ページ

MSCの**アクティビティ解析**ページを使用してカレントログファイルに記録された内容を見ることができます。この機能はデータベース利 用状況の解析、エラーや不具合の原因となった処理を探すなどの場合に役立ちます。クライアント/サーバーモードの場合、各クライアントマシンごとの操作を 検証することもできます。

**注:** データベースのデータに対して行われた操作をロールバックさせることもできます。詳細はロールバックページを参照してください。




**注:** データベースのAdministratorとDesignerのみがログファイルの検証機能にアクセスできます。

ロ グファイルに記録された操作は行として表示されます。各操作の様々な情報が列に表示されます。列のヘッダーをドラッグすることによって並び順を変えることもできます。

この情報を使用して各操作のソースとコンテキストを識別できます:

- **操作**: ログファイル中での操作の一連番号
- **アクション**: データに対して行われた操作のタイプ。この列には以下の操作のいずれかが記録されます:
  - *データファイルを開く*: データファイルを開いた
  - *データファイルを閉じる*: 開いたデータファイルを閉じた
  - *コンテキストの作成*する: 実行コンテキストを指定するプロセスを作成した
  - *コンテキストを閉じる*: プロセスを閉じた
  - *追加*: レコードを作成、格納した
  - *BLOB*を追加: BLOBフィールドにBLOBを格納した
  - *削除*: レコードを削除した
  - *更新*: レコードを更新した
  - *トランザクションの開始*: トランザクションを開始した
  - *トランザクションの受け入れ*: トランザクションを受け入れた
  - *トランザクションのキャンセル*: トランザクションをキャンセルした
- **テーブル**: レコードを追加/削除/更新、またはBLOBが属する、テーブル
- **プライマリーキー/BLOB**: それぞれのレコードのプライマリーキーのコンテンツ(プライマリーキーが複数のフィールドがから構成されているときには、値はセミコロンで分けられています)、またはオペレーションに関連したBLOBのシーケンス番号
- **プロセス**: 処理が実行された内部プロセス番号。この内部番号は処理のコンテキストに対応します。
- **サイズ**: 操作により処理されたデータのサイズ (バイト)
- **日付と時刻**: 処理が実行された日付と時刻
- **ユーザー**: 処理を実行したユーザー名。クライアント/サーバーモードではクライアントマシン名が表示されます。シングルユーザーモードではユーザーのIDが表示されます。
  4Dパスワードが有効にされていない場合、この列にはなにも記録されません。
- **値** (デフォルトでは表示されていません): レコードの追加や更新の場合、フィールドの値。値は";"で区切られます。文字形式に表現できる値のみを表示します。
- **レコード**: レコード番号

選択したデータベースのカレントログファイルの内容を更新するには**解析**をクリックします。**選択...**ボタンをクリックするとデータベースの他のログファイルを選択できます。**書き出し...**ボタンを使用してファイルの内容をテキストとして書き出せます。

## 検証ページ

このページでは、データおよび構造上の整合性を検証できます。検証は、レコードやインデックス、またデザインオブジェクト (メソッド、フォームなど) について実行できます。
この機能は検証のみをおこないます。エラーが見つかり修復が必要な場合は修復ページページを使用するよう表示されます。

**アクション**

このページには、検証機能に直接アクセスするための、次の4 つのボタンが置かれています:

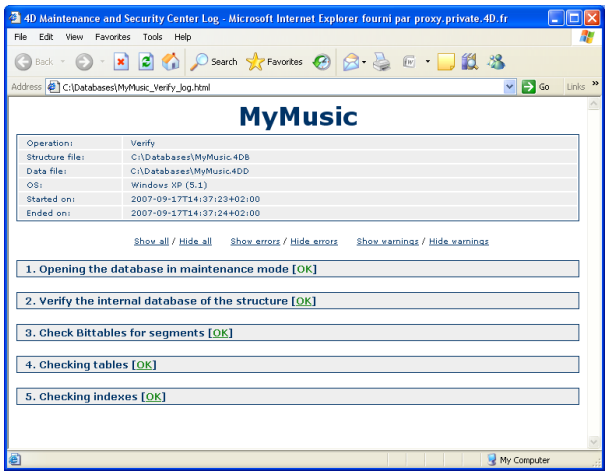
- **レコードとインデックスを検証**: 全体のデータ検証処理を開始します。
- **レコードのみを検証**: レコードのみの検証処理を開始します (インデックスは検証されません)。
- **インデックスのみを検証**: インデックスのみの検証処理を開始します (レコードは検証されません)。
  警告: このオプションではインデックスの物理的な整合性の検証のみを行います。データとのリンク検証は行いません。無効なキーを検知することはできますが、重複キーを検出することはできません (同じレコードを指す2つのインデックスキー)。このタイプのエラーは**レコードとインデックスを検証**を使用しないと検知できません。
- **アプリケーションを検証**: デザインモードで定義されたすべてのオブジェクト (テーブル、メソッド、フォームなど) の検証処理を開始します。

**注:** レコードとインデックスの検証はテーブルごとに詳細モードで行うこともできます (後述)。

**ログファイルを開く**

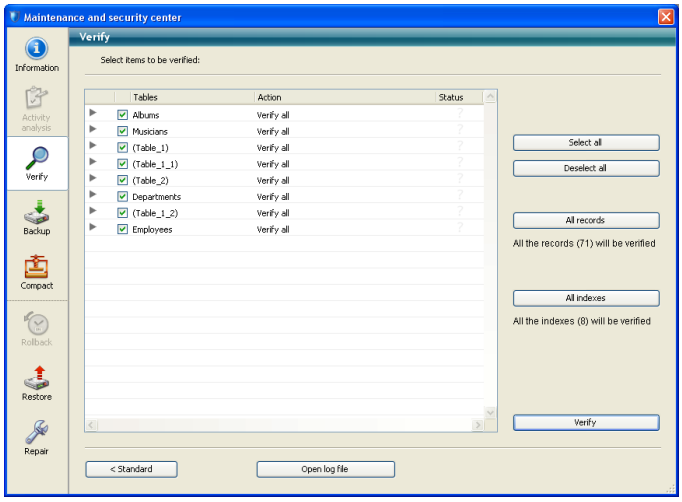
要求された検証に関係なく、4D はデータベースフォルダーにログファイルを生成します。このファイルには実行された検証の内容が記録され、エラーがあればそれも示されます。問題がない場合は[OK]が表示されます。ファイル名は"*DatabaseName_Verify_log*"でXMLおよびHTML形式で作成されます。

**ログファイルを開く**ボタンをクリックすると、4Dはマシンのデフォルトブラウザーを使用してHTMLフォーマットのファイルを開きます:




**詳細**

**詳細**ボタンは、検証するレコードおよびインデックスを選択するために使用する詳細ページを表示します。:




検証する項目を指定することにより、検証処理にかかる時間を節約できます。

リストには、データベースの全テーブルが表示されます。各テーブルに対して、検証対象をレコードやインデックスに限定できます。三角形のアイコンをクリックしてテーブルまたはインデックス付フィールドの内容を展開し、要求に応じてチェックボックスにチェックマークを入れたり外したりします。デフォルトでは、すべての項目にチェックマークが入っています。ショートカットボタン**すべて選択**、**すべての選択をはずす**、**すべてのレコード**および**すべてのインデックス**も使用できます。

テーブルの各行に対して、"アクション" カラムは実行する操作を表示します。テーブルが展開されると、"レコード" および"インデックスフィールド" の行は関連する項目の数を表示します。

ステータスカラムは、記号を使用して各項目の検証ステータスを表示します:

- ✔ 検証の結果問題はなかった
- ✘ 検証の結果問題が見つかった
- ▬ 部分的に検証が実行された
- ? 検証は実行されなかった

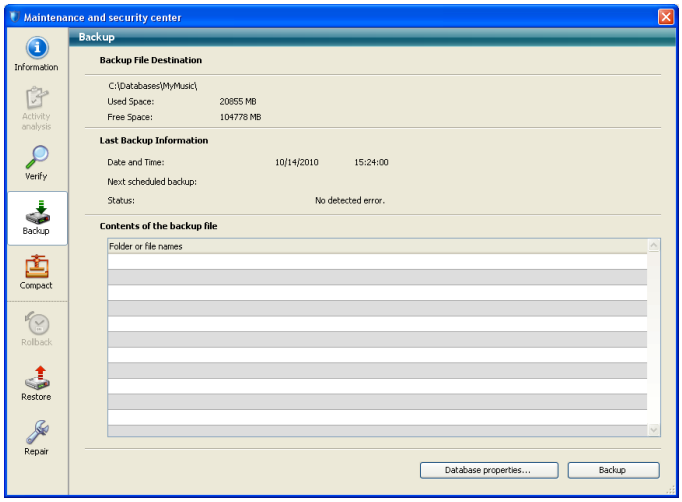
検証を開始するには**検証**をクリック、標準ページに戻る時は**< 標準**をクリックします。

**ログファイルを開く**ボタンをクリックするとマシンのデフォルトブラウザーを使用してHTML形式のログファイルを表示します (ログファイルを開く参照)。

**注:** 標準ページは、詳細ページで行われた変更はまったく考慮しません。標準ページの検証ボタンをクリックすると、すべての項目が検証されます。逆に、詳細ページで行われた設定は、セッションから次のセッションに保持されます。

## バックアップページ

MSCの**バックアップ** ページは、データベースのバックアップ 設定を表示し、手動のバックアップ作業を開始する時に使用します:




このページは以下の3つのエリアで構成されています:

- **バックアップファイルの保存先**: データベースのバックアップファイルの場所に関する情報を表示します。また、ここはバックアップディスクの空いている/ 使用されているスペースも表示します。
- **前回のバックアップの情報**: データベースで最近行われた( 自動または手動の) バックアップの日付および時間を提供します。
- **バックアップファイルの内容**: バックアップファイルに含まれるファイルおよびフォルダーをリストアップします。

**バックアップ**ボタンは、手動のバックアップを開始する時に使用します。バックアップに関する詳細はバックアップの開始を参照してください。

このページからバックアップパラメーターを変更することはできません。これを行うには**データベースプロパティ...**ボタンをクリックします。

# 圧縮ページ

このページは、データおよびストラクチャーファイルの圧縮機能にアクセスする時に使用します (4Dファイルの説明参照)。

**ファイルを圧縮する理由**

ファイルの圧縮は以下のニーズに応えるために行います:

- ファイルのサイズを減らすことと最適化: ファイルには使っていないスペースがあるかもしれません。実際、レコードやフォームなどを削除すると、それらがファイル上で占有していたスペースが空になります。4D はできる限り、こういったのスペースを再利用しますが、データのサイズは可変なため、連続的に削除や変更を行うと、必然的にプログラムにとって使用不可のスペースを作り出します。大量のデータが削除された直後についても同じことが言えます: 空のスペースはそのままファイルに残ります。
  データファイルのサイズと実際にデータに使われているスペースの比率をデータの使用率と呼びます。使用率が低すぎると、スペースが無駄なだけではなく、データベースパフォーマンスの低下につながります。圧縮は空きスペースを取り除き、データのストレージを再編成、最適化するために行います。
  "情報 "エリアにはフラグメンテーションに関するデータが要約され、必要な操作が表示されています。MSCの情報ページの データおよびストラクチャー タブにはデータベースファイルのカレントのフラグメンテーションについて詳細情報が表示されます。

**注:** 圧縮はメンテナンスモードでのみ可能です。標準モードでこの操作を実行しようとすると、警告ダイアログボックスが表示され、データベースを終了してメンテナンスモードで再起動することを知らせます。ただし、データベースによって開かれていないデータファイルを圧縮することは可能です (レコードとインデックスの圧縮参照)。

**データやストラクチャーファイルの圧縮**

データおよびストラクチャーファイルの標準 圧縮手順は全く同じです。

データまたはストラクチャーファイルの圧縮を直接開始するには、MSC ウィンドウの対応するボタンをクリックします:

データ:

ストラクチャー:

**注:** 圧縮はオリジナルファイルのコピーを伴うため、ファイルのあるディスクに十分なスペースがない場合、ボタンは使用不可になります。

この操作は、メインファイルの他、インデックスファイルもすべて圧縮します。4D はオリジナルファイルをコピーし、それをオリジナルファイルの隣に作成された**Replaced Files (Compacting)**フォルダーに置きます。操作が完了すると、圧縮ファイルは自動的にオリジナルファイルと置き換えられます。データベースは即座に操作可能になります。

**注:**

- 元のファイルのコピー先は、上級モードを使って変更できます
- 圧縮操作を複数回実行すると、毎回新しいフォルダーが作成されます。フォルダー名は、"Replaced Files (Compacting)_1 "のようになります。

**警告:** 圧縮操作は毎回オリジナルファイルのコピーを伴うため、アプリケーションフォルダーのサイズが大きくなります。アプリケーションのサイズが過剰に増加しないよう、これを考慮することが大切です (特に、4D アプリケーションがパッケージとして現れるMac OS の場合)。パッケージのサイズを小さく保つためには、パッケージ内オリジナルファイルのコピーを手動で削除することも役立ちます。

圧縮が完了すると、4D はデータベースフォルダーにログファイルを生成します。databasename_Compact_Log.xml と名付けられるこのファイルは、XMLとHTMLフォーマットで作成されます。これを使用して、実行されたすべての操作を表示できます。
ログファイルを開くをクリックすると、4D はマシンのデフォルトブラウザーを使用してHTMLフォーマットのログファイルを開きます (ログファイルを開く参照)。

**上級モード**

圧縮ページには、データおよびストラクチャーファイルの圧縮に関するオプションページにアクセスする時に使用できる**上級>**ボタンがあります:

**レコードとインデックスの圧縮**

"レコードとインデックスの圧縮"にはカレントデータファイルのパス名と、他のデータファイルを指定するために使用する **[...]** ボタンが表示されます。このボタンをクリックすると標準のファイルを開くダイアログが表示され、圧縮するデータファイルを選択することができます。開かれているストラクチャーファイルと互換性のあるデータファイルを選択しなければなりません。
このダイアログボックスを受け入れると、圧縮するファイルのパス名が更新されます。

2つめの **[...]** ボタンを使用して元のファイルのコピーの保存先を変更できます。このオプションは特に大きなデータファイルを圧縮する際に、コピー先を他のボリュームに変更するために使用します。

**レコードの強制更新**

このオプションが選択されていると、4Dは現在のストラクチャー定義に基づき、圧縮処理中に各テーブルのすべてのレコードを再保存します。このオプションが選択されていないと、4Dは単にディスク上のデータの並びを再構成するだけです。このオプションは以下のケースで有用です:

- データが入力された後にストラクチャー定義のフィールド型が変更された場合、例えば倍長整数型を実数型に変更したようなケースです。4Dでは (データを失うリストがあるにしても) まったく異なる型に変更することさえ可能です。例えば実数型をテキスト型にすることができます。
  この場合、4Dは既に入力されたデータを遡及的に変換することはしません。データはレコードがロードされ保存される際に変換されます。このオプションを使用すればデータの変換を強制できます。

- データが入力された後にテキスト、ピクチャー、BLOBの外部保存オプションが変更された場合。これは特にデータベースをv13以前から変換し、新しいオプションを変更した場合に発生します (データをデータファイル外に保存参照)。先の型変更と同様、4Dはすでに入力されたデータを遡及的に変換しません。変換を行って新しいオプションを適用するために、このオプションを選択して圧縮を行います。

- テーブルやフィールドが削除された場合。この場合、レコードの再保存を行いながら圧縮することで、この削除された領域を圧縮することができ、ファイルサイズを減らすことができます。

**注:** このオプションが選択されているとすべてのインデックスが更新されます。

**アドレステーブル圧縮**

(レコードの強制更新を選択した場合にのみ選択可能)

このオプションを使用すると圧縮中、完全にレコードのアドレステーブルを再構築します。これによりアドレステーブルのサイズが最適化されます。このオプションは主に大量のデータを作成し、そして削除したような場合に使用します。そうでない場合、最適化に明白な意味はありません。

このオプションを使用した場合、圧縮処理に時間がかかるようになり、さらにSAVE SETコマンドを使用して保存したセットなど、レコード番号に依存するものが無効になる点に留意してください。保存したセットはすべて削除するよう強く推奨します。そうでなければ不正なデータセットを使用することになります。

**注**:

- 圧縮は、ゴミ箱に入れられたテーブルのレコードも対象となります。ゴミ箱に大量のレコードがある場合、処理が遅くなる原因となります。
- このオプションを使用すると、アドレステーブルはカレントログファイルとの互換性を失います。次回データベースを起動した際、ログファイルは自動で保存され、新しいログファイルが作成されなければなりません。

**ストラクチャーファイルの圧縮**

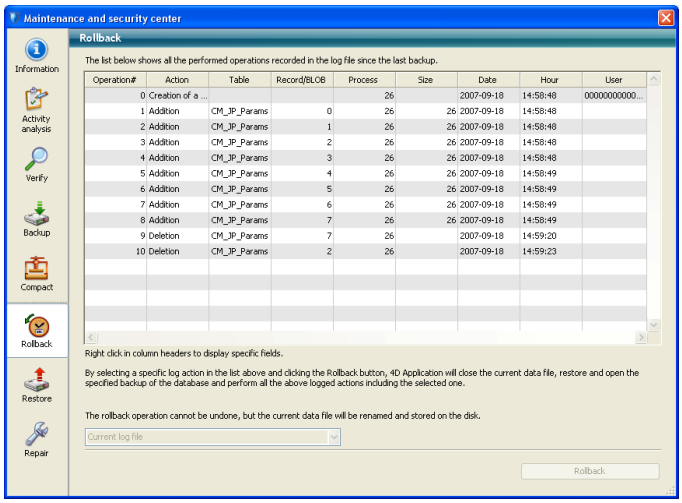
"ストラクチャーファイルを圧縮"エリアにはデータベースストラクチャーファイルのパス名と、元のファイルのコピー保存先を変更するための **[...]** ボタンが表示されます。

**圧縮** ボタンをクリックすると即座に処理が開始されます。圧縮が完了すると、4Dはデータベースフォルダーにレポートを生成します。**ログファイルを開く**ボタンをクリックすると、4Dはマシンのデフォルトブラウザーを使用してHTMLフォーマットのログを表示します (ログファイルを開く参照)。

## ロールバックページ

MSCの**ロールバック**ページはデータファイルに実行された操作にたいするロールバック機能を提供します。この機能は複数レベルに適用された取り消し機能に似ています。この機能は特に、間違ってデータベースレコードを削除した場合に便利です。

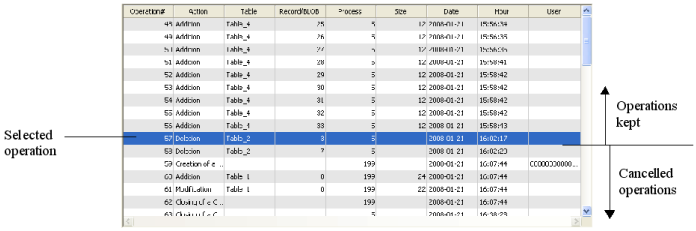
この機能はデータベースのログファイルが有効なときにのみ使用できます。




**注:** Administrator と Designer だけがログファイルを使用する機能にアクセスできます。

操作リストの内容と動作はアクティビティ解析ウィンドウのものと同じです。詳細は[アクティビティ解析ページ]を参照してください。

操作のロールバックを行うには、それよりあとのすべての操作をキャンセルする行を選択します。選択された行が保持される最後の操作になります。例えば削除をキャンセルしたい場合、その削除操作のひとつ前の行を選択します。削除操作以降の処理がすべてキャンセルされます。




ロールバックを実行するには**ロールバック**ボタンをクリックします。4Dは処理を続行してもよいか、確認してきます。

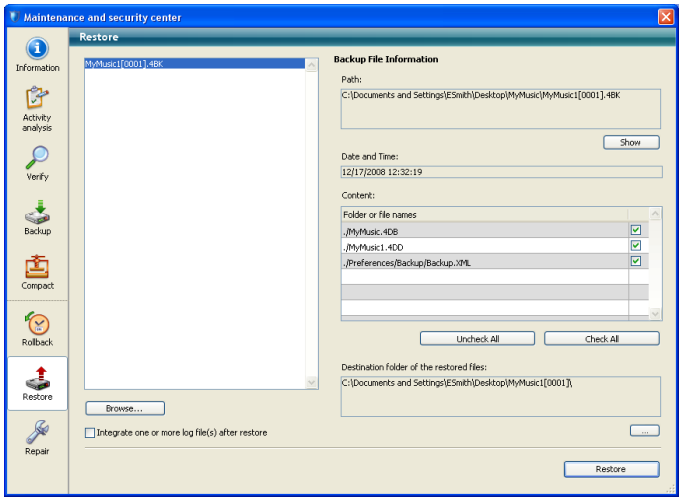
**OK**をクリックすると、データは選択された行の状態に戻ります。ウィンドウの下部にあるメニューを使用して、復旧したデータベースに対しロールバックを行う際に適用するログファイルを選択できます。この場合、アーカイブに対応するログファイルを指定しなければなりません。

ロールバックは以下のように動作します。ユーザーが**ロールバック**ボタンをクリックすると、4Dはカレントデータベースを閉じ、データベースデータの最新のバックアップから復元を行います。復元されたデータベースが開かれ、4Dはログファイル中で選択された操作までを統合します。データベースがまだ保存されていない場合、4Dは空のデータファイルを開きます。

## 復元ページ

### 手動でバックアップから復元する (MSC)

Maintenance and Security Center (MSC) の**復元**ページから、カレントデータベースのアーカイブを手動で復元できます。このページではデータベースの復元を制御するためのオプションをいくつか提供します:




ウィンドウの左側にはデータベースの既存のバックアップが表示されます。**選択...**ボタンをクリックして他の場所にあるアーカイブファイルを選択することもできます。選択したアーカイブはリストに追加されます。

このリストからバックアップファイルを選択すると、ウィンドウの右側にはこのバックアップについての説明が表示されます:

- **パス**: 選択されたバックアップファイルの完全パス名。**表示**ボタンをクリックするとシステムウィンドウでバックアップファイルが表示されます。
- **日付と時刻**: バックアップの日付と時刻
- **内容**: バックアップファイルの内容。各項目の右側にはチェックボックスがあり、復元を行うかどうか、ファイルごとに選択できます。**すべてを選択する**や**すべての選択を外す**ボタンを使用して復元するファイルの切り替えを行うこともできます。
- **復元されたファイルの保存先フォルダー**: 復元されたファイルが配置されるフォルダー。デフォルトで4Dはデータベースストラクチャーと同階層にアーカイブ名 (拡張子なし) のフォルダーを作成し、そこにファイルを復元します。個の場所を変更するには**[...]**をクリックして復元ファイルの配置場所を指定します。

#### 複数のログファイルを連続して統合する

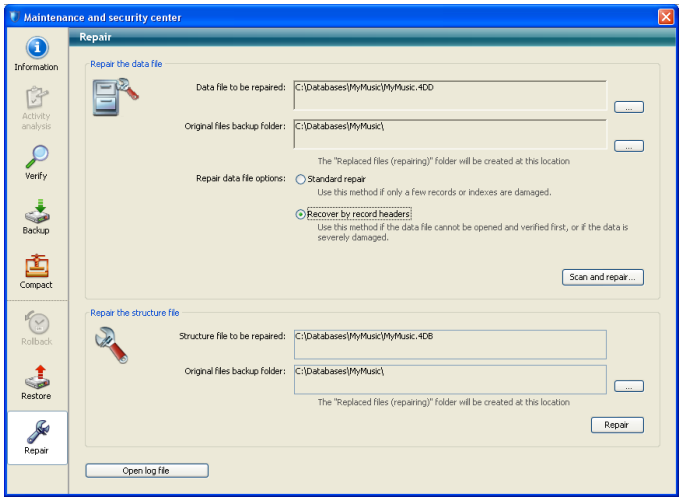
**復元後にひとつ以上のログファイルを統合**オプションを使用して、データベースに複数のログファイルを統合することができます。例えば4つのデータベースバックアップに対応する4つのログファイルアーカイブがある場合、最初のバックアップを復元して、ログアーカイブを一つずつ統合できます。これにより例えば最新のバックアップファイルを失った場合でも、データファイルを復旧することができます。

このオプションが選択されていると、4Dは復元後に標準のファイルを開くダイアログを表示します。ここで統合するログファイルを選択できます。ファイルを開くダイアログはキャンセルされるまで、統合毎に表示されます。

## 修復ページ

このページはデータやストラクチャーファイルが損傷を受けたとき、それを修復するために使用します。一般的にこれらの機能は4Dから要求を受けたときに実行します。4Dは起動時やMSCによる検証 (検証ページ参照) の結果データベースに損傷を見つけたときに修復を促します。.

**警告：** 各修復操作は元のファイルの複製を伴います。この結果アプリケーションフォルダーの容量が増大します。この点に留意することが重要です。特にMac OSではパッケージの容量が増大することを意味します。パッケージのサイズを減少させるには、作成されたコピーを削除しなければなりません。




**注：** 修復はメンテナンスモードでのみ利用できます。標準モードでこの操作を実行しようとすると、データベースが閉じられメンテナンスモードで開く旨の警告が表示されます。

**データファイル修復**

"データファイル修復"エリアにはカレントデータファイルのパス名と、他のデータファイルを指定する場合に使用する [...] ボタンがあります。このボタンをクリックすると標準のファイルを開くダイアログボックスが表示され、修復対象のデータファイルを選択できます。標準の修復を実行する場合、開かれたストラクチャーに対応するデータファイルを選択しなければなりません。レコードヘッダーによる再生を実行する場合、どのデータファイルでも選択できます。
このダイアログボックスを受け入れると、ウィンドウには修復対象のファイルのパス名が表示されます。

デフォルトで修復処理の前に元のデータファイルが複製されます。そのファイルはデータベースフォルダーの"Replaced files (repairing)"サブフォルダーに配置されます。二番目の [...] ボタンを使用して、複製の保存先を変更できます。このオプションは特にサイズの大きな元ファイルを別ボリュームに複製させたい場合に使用できます。

修復オプションは二種類、**標準の修復**と**レコードヘッダーによる再生**があります。これらについては後ほど説明します。

修復対象のデータファイルを選択したら、**修復**または**スキャンおよび修復...**ボタンをクリックすると、修復処理が開始されます (選択した修復オプションによりボタンが変化します)。レコードヘッダーによる再生を選択した場合、再生する項目を選択するためのダイアログが表示されます (レコードヘッダーによる再生参照)。

4Dは元のファイルの場所に空のデータファイルを作成します。元のファイルは、"\Replaced Files (Repairing) {日付} {時刻}"という名前のフォルダーが"元のファイルのバックアップフォルダー"に作成され (デフォルトはデータベースフォルダー)、そこに移動されます。
修復操作が完了すると、MSCの"修復"ページが表示され、修復の結果が表示されます。修復に成功していればデータベースをそのまま使い始めることができます。

**ログファイルを開く**ボタンをクリックするとマシンのデフォルトブラウザーに実行された処理の結果が表示されます。このページには実行された検証および修復の結果、そして発生したエラーが表示されます (検証に成功した項目には[OK]が表示されます)。ファイルはデータベースの**Logs**フォルダーに作成されます。ファイルはXMLおよびHTMLフォーマットで"DatabaseName_Repair_log"という名前で生成されます。

#### 標準の修復

数が多くないレコードやインデックスが損傷を受けているケースでは、標準の修復を選択します。データは圧縮および修復されます。このタイプの修復を行うためにはデータファイルとストラクチャーファイルが対応していなければなりません。

#### レコードヘッダーによる再生

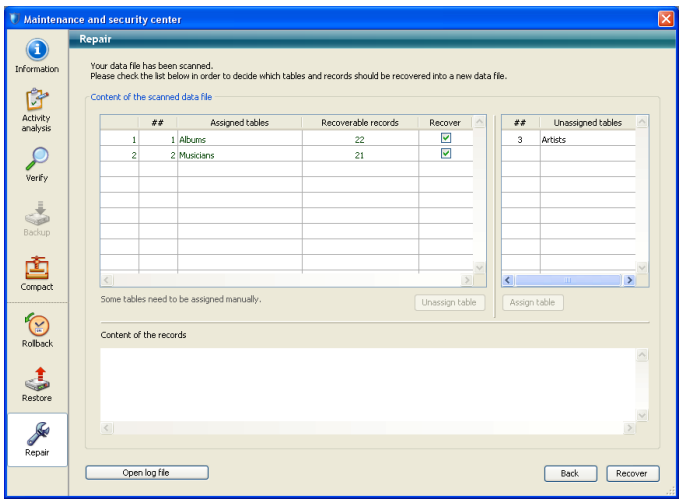
この低レベルな修復オプションは、データファイルが大きく損傷していて、バックアップからの復旧や標準の修復などの手段では回復できなかった場合にのみ使用します。

4Dのレコードはサイズが可変です。故にそのレコードをロードするためには、ディスク上のどこに格納されているか、その場所を記録しておく必要があります。そのために4Dはインデックスやアドレステーブルを経由して、レコードのアドレスにアクセスします。レコードやインデックスのみが損傷を受けている場合、通常、標準の修復を使用すれば問題は解決されます。しかしアドレステーブル自身が損傷を受けている場合、より専門的な修復作業が必要となります。これを行うために、MSCは各レコードのヘッダーに存在するマーカーを使用します。マーカーはレコード情報のサマリと比較されます。そしてアドレステーブルが再構築可能となります。

**注：**

- データベースストラクチャーのテーブルプロパティでレコードを完全に削除オプションの選択を解除していると、ヘッダーマーカーを使用した復旧によって削除したはずのレコードが復活する原因となります。
- ヘッダーによる再生は参照整合性を考慮しません。この処理を行った後、重複不可フィールドに重複する値が現れたり、NULL値を許可しないに設定したフィールドにNULL値が現れたりするかもしれません。

**スキャンおよび修復...**をクリックすると、4Dはデータファイルを完全にスキャンします。スキャンを完了すると、結果が以下のウィンドウに表示されます:



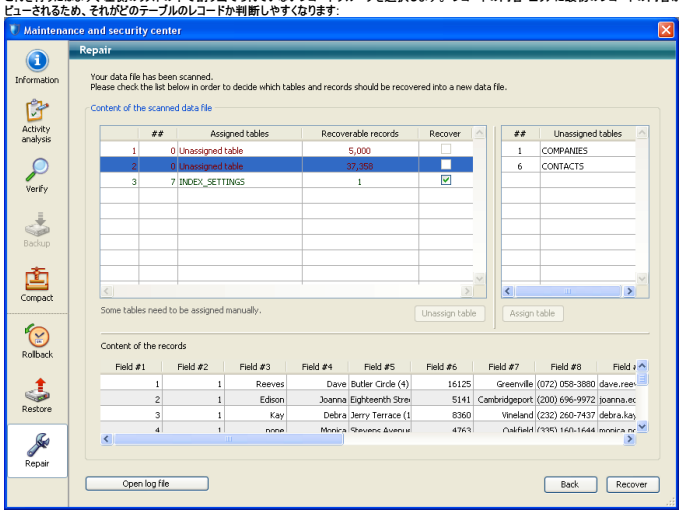

**注：** すべてのレコードおよびすべてのテーブルを対象とした場合、メインエリアのみが表示されます。

"データファイル中で見つかったレコード"エリアには2つのリストがあり、データスキャン結果の概要が表示されます。

- 左のリストにはデータファイルスキャンの情報が表示されます。各行にはデータファイル中の再生可能なレコードのグループが表示されます:
  - いちばん左の列にはレコードグループの再生順が表示されます。
  - 数列にはグループに割り当てられたテーブル番号が表示されます。
  - **割当先テーブル**列には識別されたレコードのグループに割り当てられたテーブルの名前が表示されます。割り当てられたテーブルの名前は自動で緑色で表示されます。
  - **再生可能レコード**列にはグループ中で再生可能なレコード数が表示されます。
  - **再生**列では各グループごとに、それらのレコードを再生するかどうかを指定できます。デフォルトですべてのグループが選択されています。

- 右側のリストには割り当てられなかったテーブルが表示されます。例えばどのレコードも割り当てることができなかったテーブルです。

**手動による割り当て**
アドレステーブルが損傷を受けているため、テーブルに割り当てることのできないレコードグループがある場合、それらを手動で割り当てることができます。これを行うにはまず、左側のリストの中で割り当てられていないレコードグループを選択します。"レコードの内容"エリアに最初のレコードの内容がプレビューされるため、それがどのテーブルのレコードか判断しやすくなります:




次に"割り当てられていないテーブル"リストから、グループを割り当てるテーブルを選択し、**テーブルを割り当て**ボタンをクリックします。割り当てのためにドラッグ&ドロップを使用することもできます。
結果そのレコードグループは選択したテーブルに割り当てられ、そのテーブルのレコードとして再生されます。手動で割り当てられたテーブルはリスト中黒色で表示されます。
**テーブルを割り当てない**ボタンをクリックすると、レコードグループに対するテーブルの割り当てを手動で解除できます。

**ストラクチャーファイルの修復**

"ストラクチャーファイルの修復"エリアにはデータベースストラクチャーファイルのパス名と、修復実行前に元のファイルのコピーを作成する保存先を変更するための **[...]** ボタンが表示されます。

**修復**ボタンをクリックすると、必要であればデータベースが閉じられ、再びメンテナンスモードで開かれます。

修復操作が終了すると、4Dはデータベースフォルダーに処理の結果を記録したログファイルを、XMLとHTMLフォーマットで"databasename_Repair_log"という名前で生成します。**ログファイルを開く**ボタンをクリックすると、4Dはマシンのデフォルトブラウザーを使用してHTMLフォーマットのログファイルを開きます (ログファイルを開く参照)。

# コンパイル


概要

コンパイラーウィンドウ

コンパイル設定

コンパイルの診断ツール

## 概要

お使いのデータベースをコンパイルすることができます。コンパイルとは、すべてのメソッドをマシン言語に翻訳することです。データベースをコンパイルすると、コードの整合性を調べたり、実行速度を向上させることができます。さらに、コード全体を保護することも可能です。4D でデータベース開発を行い、それをスタンドアロンアプリケーションとして配布するまでの間で、コンパイルは不可欠な作業です。

コンパイル処理はすべて自動的に行われますが、コンパイルを実行するためには4D コードをより正確に記述する必要があります。『4D ランゲージリファレンス』マニュアルのコンパイラでは、コンパイルを目的としてプログラムを作成する上でのアドバイスや特定情報が提供されています。さらに、コンパイラーはプログラムのあらゆるエラーを指摘し、エラー箇所を示すという点に留意してください。

### コンパイラーとは

コンピューターは、"0"と"1"だけで命令が記述されている装置です。この2 進数による言語は"マシン語"と呼ばれます。マシンの心臓部であるマイクロプロセッサーは、この言語しか理解することができません。高水準コンピューター言語（C、C++、Java、BASIC、4D 等）で記述されたプログラムは、コンピューターのマイクロプロセッサーが理解できるように、まずマシン語に翻訳されます。

これを行うには、2 つの方法があります:

- ステートメントの実行中に翻訳する：このプログラムはインタープリターと呼ばれます。
- プログラムの実行前にステートメント全体を翻訳する：このプログラムは、コンパイル済みと呼ばれます。

#### インタープリターモード

一連のステートメントをインタープリターを使用して実行する場合、その処理過程は次のような段階に分かれます:

- プログラムは、ステートメントをプログラム独自の言語で読み込みます。
- ステートメントをマシン語に翻訳します。
- ステートメントを実行します。

このサイクルがプログラムの各ステートメントごとに行なわれます。このタイプの実行サイクルを処理するプログラムをインタープリターと呼びます。データベースの開発過程において、4D メソッドはインタープリターモードで実行されます。

#### コンパイル済みモード

コンパイル済みのプログラムは、実行前にプログラム全体が翻訳されます。この処理により、一連のマシン語のステートメントを納めた新しいファイルが生成されます。このファイルは繰り返し使用するために保存されます。翻訳は一度だけ行なわれ、このコンパイル版のプログラムは、繰り返し実行することができます。

このフェーズでは、そのプログラムが使用されることはまったくありません。翻訳を行うプログラムは"コンパイラー"と呼ばれます。

#### コンパイル済みオブジェクト

4D のコンパイラーは、データベース内のデータベースメソッドやプロジェクトメソッド、トリガー、フォームメソッド、オブジェクトメソッドをコンパイルします。アプリケーション内にこれらの要素がひとつも存在しない場合、コンパイラーがコンパイルする対象は何もありません。

コンパイルが正常に終了すると、オリジナルのデータベースと同じ方法でコンパイル済みデータベースを使用することができます。

### データベースをコンパイルする理由

コンパイルの第一の利点は、もちろん実行速度の向上です。さらに、コンパイルに直接関係する利点が2 つあります:

- 系統的なコードチェック
- データベースとコンポーネントの保護

#### 実行速度

実行速度の向上は、コンパイル済みコードの2 つの特性によるものです。つまり、一度限りでダイレクトなコード翻訳、変数とメソッドのアドレスへのダイレクトなアクセスという2 つの特性です。

- **ダイレクトかつ確定的なコード翻訳**
  4D で作成されたメソッドのコードは、コンパイラーにより一度だけ翻訳されます。コンパイル済みデータベースを使用すると、インタープリターモードですべてのステートメントを翻訳するために要する時間が省かれます。 次に示すのは、これを説明する簡単な例です。50 回繰り返される一連のステートメントを含むループを例に取り上げています:
  ```
  For($i;1;50)
    // 一連のステートメント
  End for
  ```
  インタープリター版のデータベースでは、ループ内の各ステートメントは50 回翻訳されます。コンパイラーを使用すると、各ステートメントの翻訳フェーズがなくなるため、ループ内の各ステートメントに関して、50 回分の翻訳が省略されます。

- **変数とメソッドのアドレスへのダイレクトアクセス**
  インタープリター版のデータベースでは、名前を用いて変数にアクセスします。したがって、4D が変数の値を取得するには、その名前を使用しなければなりません。
  コンパイル済みコードの場合、コンパイラーは各変数に対してアドレスを割り当て、変数のアドレスをコードに直接書き込み、必要になる度にそのアドレスへ直接アクセスします。
  **注:**
  - ディスクへのアクセスが必要となる処理の場合、その実行速度はコンピューターと周辺装置（ドライブやハードディスク）間の伝送速度により制限を受けるため、コンパイルの効果が得られない可能性があります。
  - コメントは翻訳されないため、コンパイル後のコードには含まれません。したがって、コメントはコンパイル済みモードでの実行速度には影響しません。

#### コードをチェックする

コンパイラーは、データベースのシンタックスチェッカーとしても機能します。作成したコードを系統的にチェックして、不明確な箇所があれば知らせます。一方、4Dではメソッドの実行時にのみこの処理が行なわれます。

例えば、あるメソッドには一連の判定文と、実行対象となる一連のステートメントが含まれているものとします。テストする数が非常に多い場合は、すべてのケースを完全にテストすることはほとんどありません。この場合、テストされていないケースにシンタックスエラーが含まれていたとしても、エンドユーザがそのケースに出会うまで分かりません。

コンパイル済みデータベースを使用すると、このような問題が起きなくなります。データベースをコンパイルする際、コンパイラーはデータベース全体を細かく調べ、各ステートメントを解析します。コンパイラーはあらゆるエラーを検出し、警告やエラーメッセージを生成します。

#### アプリケーションとコンポーネントを保護する

データベースをコンパイルしたら、アプリケーションビルダーを使用してインタープリターコードを消去することができます。これを行うと、レコードアクセスを除き、デザインモードへアクセスできなくなります。アプリケーションビルダーについてはを参照してください。

コンパイル済みデータベースでは開発に関連するコマンドを使用することができません。

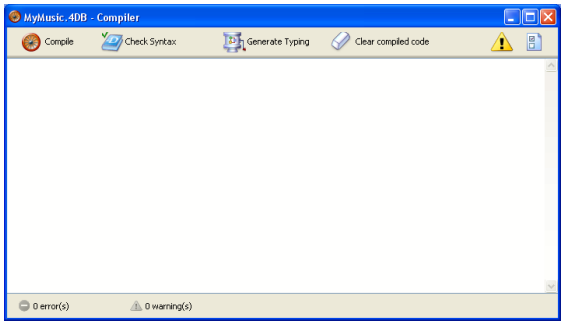
コンパイルされたコンポーネントをホストデータベースにインストールすると、エクスプローラーから共有メソッドにアクセスでき、ホストデータベースからそのメソッドを呼び出すことができます。しかしそのコードはエクスプローラーのプレビューエリアやデバッガーに表示されません。共有されないプロジェクトメソッド名はまったく表示されません。コンポーネントに関する詳細は4Dコンポーネントの開発とインストールを参照してください。

利点は以下の通りです:

- データベースのストラクチャーは、故意にしろ偶然にしろ変更されることはありません。
- メソッドは保護されます。

### 4D におけるコンパイル

4D にはコンパイラーが組み込まれています。次のダイアログボックスを使用して、データベースをコンパイルします:




**デザイン**メニュー内の**コンパイル開始**コマンドやツールバーのコンパイラーボタンに関連付けられたメニューを使用して、カレント設定に基づきコンパイル

を直接起動することもできます。

コンパイルはデータベース設定のコンパイラーページで設定されたコンパイルオプションを使用して実行されます。

データベースがコンパイルされると、**実行**メニューの**インタープリター再起動**や**コンパイル済み再起動**メニューを使用して、インタープリターモードとコンパイル済みモードを切り替えて実行できるようになります。4Dアプリケーションを終了する必要はありません (アプリケーションビルドを行ってインタプリターコードを削除している場合は除きます)。

データベースを開くダイアログボックスでも起動時にインタープリターモードとコンパイル済みモードを選択することができます (開くダイアログオプション参照)。

インタープリターモードでストラクチャーやコードを変更したら、それをコンパイルコードに反映させるために再コンパイルしなければなりません。

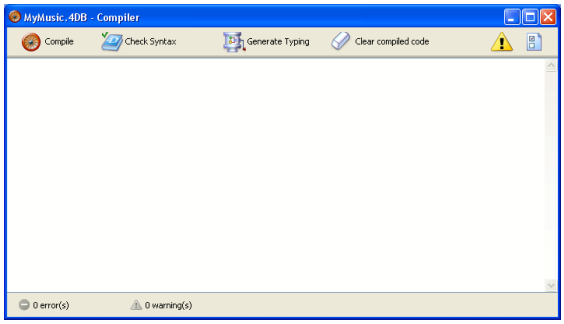
モードを変更すると、4Dは現在のモードを閉じ、新しいモードを開きます。つまりアプリケーションが閉じられ、再び開かれます。モードを移動するたびに4Dは以下のデータベースメソッドを実行します: On Exitデータベースメソッド -> On Startupデータベースメソッド

## コンパイラーウィンドウ

コンパイラーウインドウを表示するには、**デザイン**メニューの**シンタックスチェック...**コマンドを選択するか、またはデザインモードのツールバーにある**コンパイラー**ボタンをクリックします:

**注:** データベースにメソッドが1 つも存在しない場合、これらのコマンドは選択不可になります。

このウインドウを使用して、データベースのコンパイルの開始や、メソッドのシンタックスチェックを行います。さらに各ボタンを使用して、データベースの変数定義メソッドの生成や再生成、コンパイル済コードの消去、警告の表示や非表示、環境設定への直接アクセスなどの操作を行うことができます。

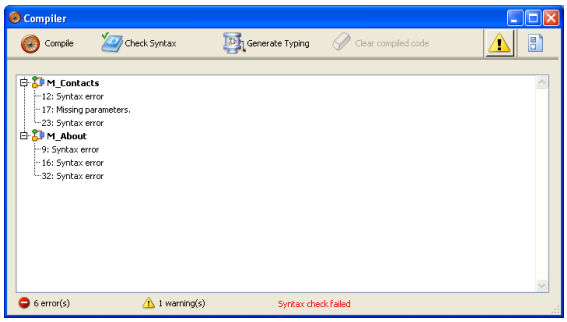



**注**: データベースのコンパイルには、適切なライセンスが必要です。ライセンスがない場合、コンパイルを実行することができません (**コンパイル開始**コマンドと**コンパイラー**ボタンは無効になります)。その場合でも、シンタックスチェックと変数定義メソッドの生成は行うことができます。

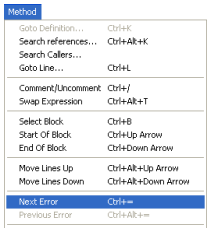
### コンパイラー

**コンパイラー**ボタンは、データベースのコンパイル処理を直ちに開始します。このボタンの動作は**デザイン**メニューの**コンパイル開始**コマンドとまったく同じです。データベースが既にコンパイルされている場合、新たにコンパイルされたコードで以前のコードが置き換えられます。
まず初めにデータベース設定のコンパイラーページ上の設定に基づき、チェックや変数定義、初期設定などの各種パスが実行されます。

エラーが検出されなければ、実際のコンパイルが開始します。エラーが検出されると、処理が中止され、問題となるメソッド名と行番号がウインドウの情報エリアに階層リスト形式で表示されます:




関係するメソッドを直接4D のメソッドエディター上で開くには、検出された各エラーをダブルクリックします。エラーを含む行が反転表示され、エラーのタイプがウインドウのシンタックスエリアに表示されます。エディターの**メソッド**メニューから**前のエラー** / **次のエラー**コマンドを選択すると、エラーが含まれる各行を移動することができます。




**注:** 初めてのコンパイルで検出されるエラーの数に辟易するかもしれませんが、気にすることはありません。ほどなく、これらのエラーが同じ原因によるものであることに気づくでしょう。例えば特定のデータベース規約に対する違反などです。常にコンパイラーでは、エラーの訂正に役立つように、正確なエラー診断が提供されます。

### シンタックスチェック

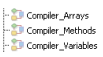
**シンタックスチェック**ボタンは、シンタックスチェックフェーズの実行を開始します。チェックが終了すると、検出されたエラーがすべて情報エリアに表示されます。エラー行をダブルクリックすると、対応するメソッドを表示することができます。

アプリケーションをコンパイルするための適切なライセンスを持たない場合は、このオプションしか使用できません。
ツールバーのコンパイラーボタンに割り当てられた**シンタックスチェック**コマンドから実行することもできます。

### 変数定義を生成する

**変数定義を生成**ボタンは、変数定義である"コンパイラーメソッド"を作成(または更新)します。コンパイラーメソッドはプロジェクトメソッドであり、すべての変数定義宣言や、プロセス配列とインタープロセス配列、およびメソッドの引数定義が集められています。コンパイラーは、これらのメソッドが存在すると、コードのコンパイル中に直接そのメソッドを利用するため、コンパイル速度が向上します。これらのメソッドが既に存在している場合、その内容が更新されます。

これらのメソッドは、4D により生成され、その名前は必ず"Compiler_"で始まります。環境設定では、5 つのコンパイラーメソッドそれぞれに対してデフォルト名を設定することができます (コンパイラーメソッド...参照)。4D により生成、管理されるコンパイラーメソッドは自動的に"非表示"属性が割り当てられます:




必要となるコンパイラーメソッド(つまり、データベース内に既存する項目のコンパイラーメソッド)だけが作成されます。

情報エリアには、メソッドの作成や更新時に検出されたエラーが示されます。エラー行をダブルクリックすると、対応するメソッドと行がメソッドエディター上に表示されます。

変数定義メソッドについては4Dランゲージリファレンスの コンパイラを参照してください。

### コンパイル済みコードをクリア

**コンパイル済みコードをクリア**ボタンを使用すると、ストラクチャファイルのコンパイル済みコードが削除されます。ボタンをクリックすると、コンパイル時に生成されたコードがすべて削除されます。圧縮を行うと、それに応じてストラクチャファイルのサイズが小さくなります (GET STYLE SHEET INFO参照)。

**実行**メニューの**コンパイル済み再起動**コマンドが無効になり、データベースを開くダイアログボックスと**開く**ボタンに割り当てられたメニュー内のコンパイルモードで開くオプションはグレー表示されます。

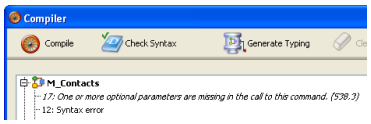
このコマンドを使用しても、生成されたコンパイラーメソッドは削除されない点に注意してください。

### 警告を表示

警告は、コンパイラーがシンタックスチェックを行う際に生成する特定のメッセージです。これらのメッセージの目的は、実行時エラーを引き起こす可能性のあるステートメントに注意を向けることです。警告によりコンパイルが中断されることはありません。警告に関する詳細は警告を参照してください。

状況や使用されるプログラミングスタイルによって、これらのメッセージの重要度は変化します。警告を表示オプションを使用すると、コンパイラーウインドウの情報エリアの警告を表示したり、または隠すことができます。

このオプションを選択すると、ウインドウには他のエラータイプの後に警告（存在する場合）が表示されます:




警告をダブルクリックすると、対応するメソッドが開かれます。

#### コンパイル時に警告を無効にする

コンパイル時に特定の警告を選択的に無効にすることができます。これを行うには、4Dメソッドのコード内に次のコードを挿入します:

```
//%W-<warning number>
```

警告番号は、コンパイルエラー一覧の各メッセージの最後に示されています。例えば、次の警告を無効にしたいものとします:

*1: 配列定義コマンド内にポインタが存在します (518.5)*

この場合、4D メソッド（できればCOMPILER_xxx メソッド）に次のコメントを記述します:

```
//%W-518.5
```

### コンパイル設定への直接アクセス

ボタンを使用してデータベース設定のコンパイラーページページを開くことができます。このダイアログボックスで設定した内容は、開いているデータベースにのみ適用されます。

# コンパイル設定

### コンパイルオプション

このエリアにはコンパイル処理中に使用される一般的なオプションがグループ化されています。

- **範囲チェック**: 範囲チェックの有効/無効を設定します。範囲チェックは追加の検証であり、実行時のアプリケーションオブジェクトの状態に基づき、コードをチェックします。範囲チェックに関する詳細は範囲チェックを参照してください。
- **Symbolファイルを生成**: 変数とその型のリスト、およびメソッドの引数や戻り値の型のリストが記述されたドキュメントを生成するために使用します。
  ファイルはデータベースストラクチャーと同階層に*DatabaseName_symbols.txt*という名前で作成されます。
  シンボルファイルに関する詳細はシンボルファイルを参照してください。
- **エラーファイルを生成**: シンタックスチェック時にエラーファイルを生成するために使用します。一般的なエラー、特定のコード行のエラー、および警告などが含まれます。
  コンパイラーが検知したすべてのエラーは4Dのメソッドメニューからアクセスできます。エラーファイルを生成し、他のマシンに転送することが有効な場合もあります。特に複数の開発者がチームで開発を行っている場合などです。
  エラーファイルはXML形式で生成されるため、内容を自動で解析することが容易になっています。このファイルを使用してカスタマイズされたエラー表示インターフェースを作成することも可能です。エラーファイルは*DatabaseName_errors.xml*という名称で、ストr買う茶ーファイルと同階層に作成されます。
  エラーファイルに関する詳細はエラーファイルを参照してください。
- **64-bit プロセッサー用にもコンパイルする**: アプリケーションを (32-bitコンパイルに加え) 4D 64-bit版用にコンパイルするために使用します。以下の点に留意してください:
  - バージョン12.1より64-bitバージョンの4D Serverが提供されています。
  - 非Unicodeモードのアプリケーションを64-bitにコンパイルすることはできません。
  - 64-bit版の4Dでアプリケーションを使用する場合にこのオプションを使用します。OSが64-bitであるかどうかは関係ありません。
- **ローカル変数初期化**: メソッドの先頭でのローカル変数初期化モードを設定するために使用します:
  - **ゼロにする**: 変数はデフォルトで0に設定されます (文字型の場合空の文字列、数値型の場合0等)
  - **ランダム値にする**: コンパイラーは無意味な値を割り当てます。この値は常に同じであり、例えば倍長整数の場合1919382119、文字の場合"rgrg"、ブールの場合True等になります。このオプションを使用すると、初期化を忘れているローカル変数を探すのが容易になります。
  - **いいえ**: コンパイラーはローカル変数を初期化しません。この設定を使用するとアプリケーションの実行時間を短縮できます。ただし開発者が正しくローカル変数を初期化しなければなりません。
- **コンパイルパス**: コンパイラーの実行周期数を設定するために使用します。
  - **すべて定義させる**: コンパイルを行う際に可能なステージをすべて行います。
  - **ローカル変数のみ自動定義させる**: プロセスおよびインタープロセス変数の型を決定する処理は行われません。このオプションを選択する場合、すべてのプロセス変数とインタープロセス変数を開発者自身が宣言するか、自動コンパイラーメソッド生成機能を使用しなければなりません。
  - **自動変数定義は行わない**: ローカル、プロセスおよびインタープロセス変数の型を決定する処理は行われません。このオプションを千tかうする場合、すべての変数が明示的に宣言されていなければなりません。

### デフォルトの型指定

このエリアでは曖昧なアプリケーションオブジェクトのデフォルト型を設定します。

- **数値**: 実数または倍長整数から選択できます。アプリケーション中に型指定ディレクティブが書かれている場合、そちらが優先されます。倍長整数を指定することでアプリケーションを最適化できます。
- **ボタン**: 実数または倍長整数から選択できます。アプリケーション中に型指定ディレクティブが書かれている場合、そちらが優先されます。標準ボタンの他以下のオブジェクトが対象となります: チェックボックス、3Dチェックボックス、ハイライトボタン、非表示ボタン、3Dボタン、ピクチャーボタン、ボタングリッド、ラジオボタン、3Dラジオボタン、ピクチャーラジオボタン、ピクチャーポップアップメニュー、階層ポップアップメニュー、ポップアップ/ドロップダウンリスト。

### コンパイラーメソッド...

このエリアでは、コンパイラーが自動で生成するコンパイラーメソッドの名前を設定できます。これらのメソッドはすべての変数の型定義をグループ化 (プロセス変数、インタープロセス変数、プロセス配列、インタープロセス配列、メソッド) します。これらのメソッドを生成するには、コンパイラーウィンドウの変数定義を生成ボタンを使用します。詳細はコンパイラーウィンドウを参照してください。

最大5つのコンパイラーメソッドが生成されます。コンパイラーメソッドはアプリケーションに対応するオブジェクトが存在する場合のみ作成されます:

- **変数**: プロセス変数定義がグループ化されます。
- **インタープロセス変数**: インタープロセス変数定義がグループ化されます。
- **配列**: プロセス配列定義がグループ化されます。
- **インタープロセス配列**: インタープロセス配列定義がグループ化されます。
- **メソッド**: メソッド引数として使用されるローカル変数定義がグループ化されます (例: C_INTEGER(mymethod;$1))。

それぞれの対応するエリアで、作成されるメソッドの名前を編集できます。
常に"Compiler_"という接頭辞が付きます。これは変更できません。各メソッド名はユニークでなければならず、接頭辞を含み31文字以下でなければなりません。拡張文字 (アクセント文字や記号文字等) やスペースは使用できません。

## コンパイルの診断ツール

データベースの分析や訂正を行うために、3種類のツールが用意されています:

- 実際の解析用ツールはシンボルファイルにより提供されます。このテーブルを使用して、変数を素早く見つけることができます。このツールは、コンパイラーにより報告されたエラーメッセージを解析する上で非常に役立ちます。
- 訂正用ツールはエラーファイルにより提供され、このファイルはテキストファイルとして使用することができます。
- 実行ツールまたは範囲チェックは、アプリケーションの整合性と信頼性を監視するために、別のツールとして提供されます。

**注:** 自動コンパイラーメソッドにより、変数定義用の重要なツールも提供されています。 — 変数定義を生成するを参照してください。

### シンボルファイル

シンボルファイルはテキストタイプのドキュメントであり、その長さはデータベースサイズによって異なります。デフォルトでは、コンパイル時にこのファイルは生成されません。このファイルを生成するには、データベース設定の該当するオプションを選択しなければなりません (コンパイルオプション参照)。. When it is このファイルが生成されると、データベースのストラクチャが格納されているフォルダー内に配置され *DatabaseName_symbols.txt* のような名前が付けられます

テキストエディターでシンボルファイルを開くと、次のように表示されます:

```
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
```

ヘッダーにはデータベースの名前、ドキュメントの作成日付と時刻が表示されます。このドキュメントは4 つの部分に分かれています:

- インタープロセス変数のリスト
- プロセス変数のリスト
- メソッド内のローカル変数のリスト
- プロジェクトメソッドとデータベースメソッドのリスト、およびそのパラメーター(該当する場合)。

#### プロセス変数とインタープロセス変数のリスト

これら2 つのリストは、4 つのカラムに分かれています:

- 最初のカラムには、データベースで使用されるプロセス変数、インタープロセス変数、および配列の名前が納められます。変数は文字順に表示されます。
- 2 番目のカラムには、その変数のタイプが納められます。各タイプは、コンパイラー命令コマンドにより設定されるか、または変数の使われ方に基づいてコンパイラーが判断します。変数のタイプが特定できない場合、このカラムは空欄になります。
- 3 番目のカラムには、変数が配列の場合に、その次元数が表示されます。
- 4 番目のカラムには、コンパイラーが変数のタイプを決定したコンテキストへの参照が格納されます。変数が複数のコンテキストで使用されている場合は、コンパイラーが変数タイプを決定する際に使用したコンテキストが表示されます。
  - 変数がデータベースメソッド内で検出された場合、(M)* に続けて4D で定義されたデータベースメソッド名が表示されます。
  - 変数がプロジェクトメソッド内で検出された場合、(M)に続けて4D で定義されたメソッド名が表示されます。
  - 変数がトリガー(テーブルメソッド)内で検出された場合、(TM)に続けてテーブル名が表示されます。
  - 変数がフォームメソッド内で検出された場合、テーブル名と(FM)に続けてフォーム名が表示されます。
  - 変数がオブジェクトメソッド内で検出された場合、フォーム名、テーブル名、(OM)に続けてオブジェクトメソッド名が表示されます。
  - 変数がフォーム上のオブジェクトであり、プロジェクトメソッド、フォームメソッド、オブジェクトメソッド、トリガーのいずれでも使用されていない場合は、(F)に続けてそのオブジェクトが使用されるフォーム名が表示されます。

  各リストの最後には、プロセス変数とインタープロセス変数のサイズがバイト単位で表示されます。

**注:** コンパイル時に、コンパイラーは特定のプロセス変数が使用されているプロセスを判別できません。プロセス変数には、各プロセスごとに異なる値が格納されている可能性があります。そのため、新規プロセスが開始されるたびに、すべてのプロセス変数が意図的に複製されます。したがって、メモリ上でこれらのプロセス変数が占める容量に注意することをお勧めします。また、プロセス変数に必要な容量は、プロセスのスタックサイズとは関係ないという点に留意してください。

#### ローカル変数のリスト

ローカル変数のリストは、データベースメソッド、プロジェクトメソッド、トリガー(テーブルメソッド)、フォームメソッド、オブジェクトメソッドごとに、4Dと同じ順番で並べ替えられています。

このリストは、3 つのカラムに分かれています:

- 最初のカラムには、メソッドで使用されるローカル変数のリストが納められます。
- 2 番目のカラムには、その変数のタイプが納められます。
- 3 番目のカラムには、変数が配列の場合に、その次元数が表示されます。

#### メソッドの全リスト

ファイルの最後には、データベースメソッドとプロジェクトメソッドの全リストが納められ、それぞれの引数や戻り値のデータタイプも示されます。
この情報は、次の形式で示されます:

メソッド名(引数のデータタイプ): 戻り値のデータタイプ

### エラーファイル

データベース設定にあるオプションを使用して、コンパイル時にエラーファイルを生成するかどうかを選択することができます (コンパイルオプション参照)。エラーファイルを生成する場合、そのファイルには*DatabaseName_errors.xml*という名前が自動的に付けられ、データベースのストラクチャーファイルと同階層に保存されます。

コンパイラーウインドウからエラーに直接アクセスすることができますが、特にクライアント/サーバー環境で複数の開発者が共に作業を行うような体制であれば、あるマシンから別のマシンへ送信できるエラーファイルがあると便利です。エラーファイルは、その内容を自動的に解析しやすいようにXML フォーマットで生成されます。また、エラー表示用に独自のインターフェースを作成することもできます。

エラーファイルの長さは、コンパイラーにより生成されるエラーと警告の数により変わります。テキストエディターでエラーファイルを開くと、次のように表示されます:

```
<?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>
```

エラーファイルの構造は次の通りです:

- ファイルの一番上にはエラーと警告のリストがあり、メソッドごと、そして4Dで作成された順序で並べられます。
- **** 全般的なエラー****セクションには、タイプ定義が行えないものと識別が不明確なものがすべて集められます。これらのエラーと警告は、次の形式で表示されます:
  - 1 番目は、メソッドにおける行番号(0 は全般的なエラーを表わす)
  - 2 番目の"warning"属性は、検出された異常が警告であるのか(warning="true")、あるいはエラーであるのか(warning="false")を表わす
  - 3 番目には、エラーを解説する診断を表示

データベースに全般的なエラーが存在しない場合、そのファイルには“全般的なエラー”セクションがありません。

エラーファイルには、次の3 つのタイプのメッセージが含まれます:

- 特定の行に関連するエラー
- 全般的なエラー
- 警告

#### 特定の行に関連するエラー

これらのエラーは、その説明とともにコンテキスト（エラーが見つかった行）内に表示されます。コンパイラーは、データタイプやシンタックスに関する矛盾を表現式で見つけると、このタイプのエラーをレポートします。

コンパイラーウインドウでは、検出された各エラーをダブルクリックすると、該当するメソッドが直接4D のメソッドエディターで開かれ、エラーを含む行が反転表示されます。

シンタックス/タイプ定義の診断エラーの一覧は、4D ランゲージリファレンスマニュアルのエラーメッセージを参照してください。

#### 全般的なエラー

これらのエラーは、データベースのコンパイルを不可能にします。コンパイラーが全般的なエラーを生成するケースは、次の2 つです:

- プロセス変数のデータタイプが決定できない。
- 異なる2 つのオブジェクトが同じ名前である。

全般的なエラーは特定のメソッドに関連していないため、このような名前が付けられています。最初のケースの場合、コンパイラーはデータベースのいずれの箇所でも、指定されたタイプ定義を実行できません。2 番目のケースでは、いずれのオブジェクトに特定の名前を割り当てるべきかを決定できません。

全般的なエラーの一覧は4D ランゲージリファレンスマニュアルのエラーメッセージを参照してください。

#### 警告

警告はエラーではありません。警告により、データベースがコンパイルできなくなることはありません。これは、エラーになる可能性のあるコードを示すだけです。

コンパイラーウインドウにおいて、警告はイタリック体で表示されます。それぞれの警告をダブルクリックすると、該当するメソッドが直接4D のメソッドエディターで開かれ、その警告に関係する行が反転表示されます。警告の一覧は4D ランゲージリファレンスマニュアルの警告メッセージを参照してください。

特定の警告を無効にすることができます (コンパイル時に警告を無効にする参照)。

### 範囲チェック

データベース設定において、範囲チェックはデフォルトで選択されています (コンパイルオプション参照)。

他のオプションはすべてコンパイル時に機能しますが、範囲チェックはコンパイル済みデータベースの実行時にチェックを開始します。つまり、範囲チェックのメッセージは、データベースの実行中にのみ表示されます。

範囲チェックは補助的な検証を行い、論理上および構文上の矛盾点を探します。通常、これはコンパイラーの役割です。範囲チェックを行う際に、コンパイラーは“リクエストされた内容を考慮すると、得ようとしている結果は意外なものでしょうか？”という質問を投げかけます。範囲チェックは“現場”のコントローラのようなものであり、一定の時点でデータベース内のオブジェクトのステータスを評価します。

範囲チェックは次のように機能します。例えば、配列MyArrayをテキストとして定義します。MyArrayの要素数は、カレントメソッドに応じて変わるものとします。If you MyArray の5 番目の要素に“Hello”という値を代入したい場合、以下のように記述します:

```
MyArray{5}:="Hello"
```

この時点で、MyArray に5 つ以上の要素が存在すれば何も問題ありません。代入は正常に行われます。しかし、この時点でMyArray の要素数が5 つ未満であれば、代入は無効になります。

このような状況は、メソッドが実行されることを前提としているため、コンパイル時に検出することができません。コンパイラーは、このメソッドが呼び出される状況は分かりません。データベースの使用時に実際に行われる操作を監視できる方法は、範囲チェックだけです。前述の例題では、コンパイラーは4D を使用して実行時エラーを表示します。配列、ポインター、文字列を処理する場合に、範囲チェックが特に役立つ理由は、すぐにお分かりでしょう。

範囲チェックを要求した場合に、コンパイラーから生成されるメッセージは4Dランゲージリファレンスマニュアルの範囲チェックメッセージに記載されています。

#### ローカルで範囲チェックを無効にする

範囲チェックが有効な場合でも、コード内で間違いがないと思われる箇所に対して範囲チェックを適用したくないときもあります。具体的に言うと、かなりの回数繰り返されるループに関し、旧式のマシン上でコンパイル済みデータベースを実行すると、範囲チェックにより処理速度が著しく低下するおそれがあります。関連するコードに誤りがなく、システムエラーを引き起こさないことが確実であれば、範囲チェックをローカル上で無効にすることができます。

これを行うには、範囲チェックから外すコードを特殊なコメントである“%R-”と“%R+”で囲みます。//%R- コメントは範囲チェックを無効にし、//%R+ はそれを再び有効にします:

```
... //範囲チェックは有効
 // %R-
... //ここに範囲チェックから外すコードを記述する
 // %R+
... //これ以降のメソッドでは、範囲チェックが再び有効
```

**注:** このメカニズムは、範囲チェックが有効である場合にのみ機能します。

### 異常の診断

例えば、データベースの実行中に異常に気付いたものとします。その問題の原因について推測する前に、コンパイラーより提供される手がかりを思い出してください。

考えられる異常としては、次のようなものがあります:

- 4D から独自のエラーメッセージが表示される
  可能であれば、4D の指示に従ってデータベース内のエラーを修正します。指示が漠然としすぎている場合は、範囲チェックを必ず指定してデータベースを再度コンパイルします。データベースをテストし直すと、4D メッセージが表示された場所に、より情報量の多いメッセージがコンパイラーから表示されます。
- コンパイル済みデータベースとインタープリター版データベースの動作が一致しない場合
  シンタックスチェックの結果生成される警告メッセージを詳しく見てください。
- 数値変数や文字列変数が予期しない値を返す場合
  データベース設定にあるデフォルトの変数タイプオプションを確認してシンボルファイルを調べ、すべての変数が適切に型定義されているかどうかをチェックしてください。
- データベースがインタープリターモードでは動作するが、コンパイル済みモードではシステムクラッシュが発生する場合
  範囲チェックオプションを使用してデータベースをコンパイルしたかどうかを確認し、コンパイル済みデータベースがコンパイル時に使われたものと同じプラグインを使用しているかどうかを調べます。

# アプリケーションの仕上げと展開


アプリケーション展開の原則

アプリケーションビルダー

コンパイル済みストラクチャーページ

アプリケーションページ

クライアント/サーバーページ

プラグイン & コンポーネントページ

ライセンスページ

サーバーまたはシングルユーザー向けアプリの自動アップデート

スタンドアロンアプリケーションアイコンのカスタマイズ

## アプリケーション展開の原則

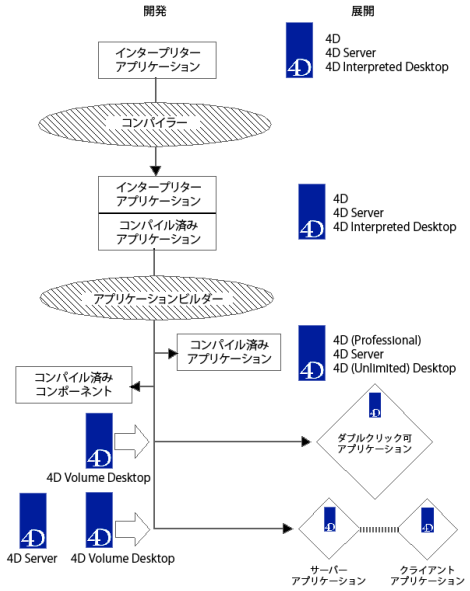
4Dにはアプリケーションビルダーが統合されています。このビルダーを使用すればコンパイルされた4Dアプリケーションの展開を簡易化することがで きます。OSごとに異なる特定の処理 (特にMac OS用のパッケージ作成) を自動で処理し、クライアント/サーバーアプリケーションの展開が容易になります。

アプリケーションビルダーでは以下のことを行えます:

- インタープリターコードを含まないコンパイル済みアプリケーションのビルド
- ダブルクリックで起動可能なスタンドアロンアプリケーションのビルド (4D Volume Desktopを組み込んだ4Dアプリケーション)
- XML形式のプロジェクトファイルに定義を行い、同じコンパイル済みストラクチャーから異なるアプリケーションのビルド
- クライアント/サーバーアプリケーションのビルド
- クライアントとサーバーの自動更新機能を備えたクライアント/サーバーアプリケーションのビルド

**4D Serverに関する注記:** アプリケーションをビルドする際には4Dシングルユーザーバージョンを使用しなければなりません。

可能な4Dアプリケーションの展開オプションは下図の通りです:

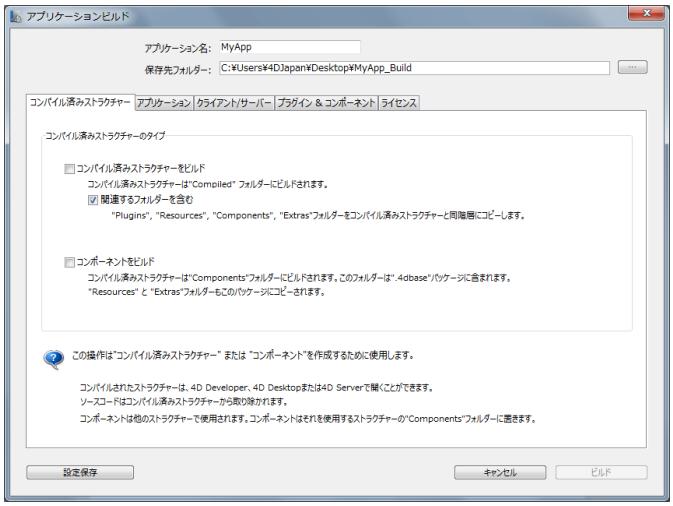


## アプリケーションビルダー

アプリケーションのビルドはアプリケーションビルドウィンドウで行います。このウィンドウを開くには4Dの**デザイン**メニューから**アプリケーションビルド...**を選択します。

ビルドを行う前にアプリケーションがコンパイルされていなければなりません。まだコンパイルされていないアプリケーションでこのメニューコマンドを選択する、あるいはコンパイルした後にコードが変更されていると、データベースを (再) コンパイルしなければならない旨の警告ダイアログが表示されます。

アプリケーションビルドウィンドウには複数のページがあり、タブを使用してページを移動できます:




- コンパイル済みストラクチャーページではコンパイルコードのみを含むストラクチャーやコンポーネントのビルド設定を行います。
- アプリケーションページではシングルユーザーアプリケーションのビルド設定を行います。
- クライアント/サーバーページではクライアント/サーバーアプリケーションのビルド設定を行います。
- プラグイン & コンポーネントページではアプリケーションに統合するプラグインやコンポーネントを選択します。
- ライセンスページでは使用するシリアル番号の選択を行います。

同時に異なるビルドパラメーター (シングルユーザーとクライアント/サーバー) の設定を行うことができます。これを行うためには各ページで対応するチェックボックスを選択しなければなりません。

**ビルド** ボタンをクリックすると選択したターゲットアプリケーションがビルドされます。このボタンがクリックされると4Dは実行中のフェーズを表示する進捗バーを表示します。有効な設定が行われれば、設定内容はアプリケーションプロジェクトに保存されます (参照)。

**設定保存** ボタンをクリックすれば、アプリケーションビルドを実行しなくてもビルドパラメーターを保存できます。

エラーが発生すると生成されたファイルは削除され、中断の理由を提供する警告ダイアログが表示されます。

### アプリケーション名と保存先フォルダー

アプリケーションビルドウィンドウの上部では、生成するアプリケーションの名前と保存先を指定します。

**アプリケーション名** エリアにはデフォルトでストラクチャーファイル名が入力されています。この欄の名前が生成されるファイル (コンパイル済みストラクチャー、コンポーネント、シングルユーザーアプリケーション、クライアント/サーバーアプリケーション) で使用されます。ビルドされるアプリケーションに基づき4Dは自動で必要な拡張子 (.4dc、.exe、server等) を付加します。
デフォルト名から変更を行わない限り、ストラクチャーファイル名を変更すればそれがアプリケーション名に反映されます。アプリケーション名を変更すると、そのアプリケーションのビルド時には指定した名前がデフォルトで使用されるようになります。入力する名前に拡張子を付けてはいけません。さらにOSがパス名として禁止する文字を含めてもいけません (Windowsでは".?!"、Mac OSでは":")。

**保存先フォルダー** エリアでは生成されるファイルの保存先を指定します。4Dはデフォルトでカレントデータベースフォルダー (Mac OSの場合パッケージ) と同階層の"*StructureName_Build*"を指定します。
保存先フォルダーを変更するにはこのエリアの右にある選択ボタン**[...]**をクリックします。フォルダー選択ダイアログボックスが表示されるので、新しい保存先を選択します。このダイアログボックスを受け入れると、フォルダーの完全パス名が入力されます。このアプリケーションのビルド時には以降このパスがデフォルトで使用されます。

ビルド実行時に4Dは自動で中間フォルダー (要求されたビルドリクエストに応じて"Compiled Database"、"Components"、"Final Application"、"Client Application"、"Server Application") を作成します。このため同時に複数タイプのアプリケーションをビルドすることができます。

### パラメーターのXMLキー

このウィンドウで設定された各パラメーターは"BuildApp.XML"という名称のアプリケーションプロジェクトファイルに格納されます。このファイルはデータベースの**Preferences**内、**BuildApp**サブフォルダーに配置されます。

ダイアログボックスが初めて表示されるときにはデフォルトパラメーターが使用されます。**ビルド**や**設定保存**ボタンをクリックすると、プロジェクトファイルの内容が更新されます。

あるデータベースについて内容の異なる複数のXMLプロジェクトファイルを定義し、BUILD APPLICATIONコマンドでそれらを使い分けることができます。

XMLキーを使用すると、アプリケーションビルドダイアログに表示される以上の設定を行うことができます。例えば:

- サーバーのIPアドレスやポート番号
- クライアントアプリケーションとサーバーアプリケーションのバージョン番号互換性 (例えばクライアントバージョン1.1から1.3がサーバーバージョン1.3に接続可能など)
- データファイルのパス名 (これにより、ユーザーマシン上でデータファイルの場所を指定する必要がなくなります)
- Windows 環境下において権限を引き上げることによって、組み込みアプリの保護されたシステムロケーションへの自動インストールを可能にします。

詳細は専用のドキュメント 4D *XML Keys-BuildApplication* を参照してください。

### ログファイル

アプリケーションを最初にビルドする際、4DはXML形式のログファイルを生成します。ログファイルは各アプリケーションプロジェクトごとに生成されます。

ログファイルは"*ProjectName*.log.xml" (*ProjectName* はアプリケーションプロジェクト名、デフォルトでBuildApp) という名称で、アプリケーション**Preferences**フォルダー内**BuildApp**サブフォルダーに配置されます。

ログファイルには各ビルド毎に以下の情報が書き込まれます:

- ターゲットビルドの開始と終了
- 生成されたファイルの名称とフルパス
- ビルドの日付と時刻
- 発生したエラー

## コンパイル済みストラクチャーページ

**コンパイル済みストラクチャー**ページでは標準のコンパイル済みストラクチャーファイルやコンパイル済みコンポーネントをビルドできます。

どちらを選択しても生成されるストラクチャーファイルは同じものです。生成されるフォルダーの構造だけが異なります。

ビルド時に4Dは生成したファイルを格納するためのサブフォルダーを自動で作成します。どちらのオプションを選択しても、カレントのアプリケーションは変更されません。ディスク上にコピーが作成されます。

コンパイル済みストラクチャーファイルに32-bitと64-bit両方のコードが含まれる場合 (コンパイルオプション参照)、対応するバージョンのコンパイル済みデータベースがビルドされます。

このウィンドウでオプションを設定したら、**ビルド**ボタンをクリックすると、指定された場所にファイルが生成されます。

### コンパイル済みストラクチャーをビルド

このオプションはコンパイル済みコードだけを含むアプリケーションをビルドします。"アプリケーション名"エリアに"MyDatabase"と指定した場合、以下が生成されます:

- MyDatabase.4dc (コンパイル済みストラクチャーファイル)
- MyDatabase.4DIndy (ストラクチャーインデックスファイル)

適切なライセンスをもつ4D、4D Server、4D Desktopが.4dcファイルを開けます。ビルド中、4Dはストラクチャーからインタープリターコードを削除します。このファイルを使用してデザインモードにアクセスすることはできません。

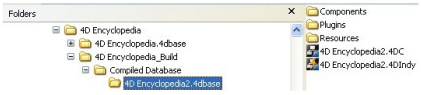
コンパイル済みアプリケーションは保存先フォルダー内の"Compiled Database"サブフォルダー内、"MyDatabase.4dbase"タイプのフォルダー (このタイプのフォルダーはMac OS Xで特別なプロパティを持ちます (.4dbase拡張子参照)) に配置されます。

**警告:** コンパイル済みアプリケーションをビルドする際に、4Dは"Compiled Database"フォルダーの内容を置換します。以前のビルド内容を保持しておきたい場合、ファイル名を変更するか移動しなければなりません。

#### 関連するフォルダーを含む

このオプションを選択すると、アプリケーションに関連するフォルダー ("Plugins"、"Resources"、"Components") がコンパイル済みアプリケーションにコピーされます。これらのフォルダーに関する情報は データベースアーキテクチャーを参照してください。

このオプションが有効になっていると、コンパイル済みストラクチャーをすぐに使い始められる状態でビルドできます。このオプションが選択されている場合の典型的な構造は以下の通りです:




### コンポーネントをビルド

このオプションを使用してストラクチャーからコンパイル済みコンポーネントをビルドできます。

コンポーネントは特定の機能を実装した4Dの標準ストラクチャーファイルです。コンポーネントをビルドして他の4Dデータベース (ホストデータベース) にインストールすると、ホストデータベースからその機能にアクセスできるようになります。コンポーネントに関する詳細は4Dコンポーネントの開発とインストールを参照してください。

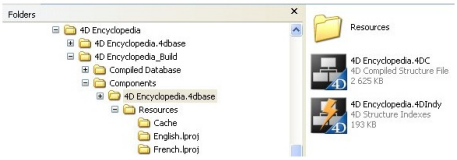
"アプリケーション名"エリアに"MyComponent"と指定した場合、以下のファイルが生成されます:

- MyComponent.4dc (コンパイル済みストラクチャーファイル)
- MyComponent.4DIndy (ストラクチャーインデックスファイル)

生成されるファイルはコンパイル済みストラクチャーと同様ですが、以下の点で異なります:

- コンポーネントは"Components"フォルダー内にフォルダーあるいはパッケージとして生成されます。
- "Resources"フォルダーは自動でコンポーネントフォルダー内にコピーされます。他方"Components"や"Plugins"フォルダーはコピーされません。コンポーネントからはプラグインや他のコンポーネントを含めることができないためです。

コンパイル済みストラクチャーの典型的な構造は以下の通りです:



## アプリケーションページ

4Dを使用して、ダブルクリックで起動可能なアプリケーションをビルドできます。このためには4Dデータベースエンジンである4D Volume Desktopと適切なライセンスが必要です。アプリケーションのビルド設定は**アプリケーション**ページで行います。

ダブルクリックで起動可能な4Dコンパイル済みアプリケーションは"スタンドアロンアプリケーションをビルド"オプションを選択するとビルドできます。Mac OSではソフトウェアパッケージが作成されます。

この処理はコンパイル済みストラクチャーファイルと4D Volume Desktopを統合します。4D Volume Desktopが提供する機能はライセンスページページで指定する、インストールされるライセンス情報に基づきます。この点についての詳細な情報は、4Dのサイト(http://www.4d.com/)と、セールスドキュメンテーションを参照して下さい。

ランゲージコマンドのシークエンスを使用することによって、シングルユーザー向け組み込みアプリケーションのアップデートを自動化することが可能です(サーバーまたはシングルユーザー向けアプリの自動アップデート を参照して下さい)。

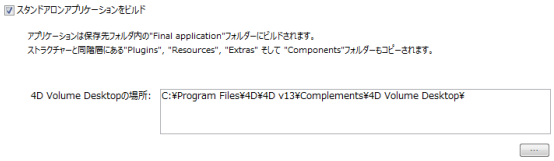
### 4D Volume Desktopの場所

ダブルクリックで起動されるアプリケーションをビルドするためには、まず4D Volume Desktopフォルダーの場所を指定しなければなりません。このエリアにフォルダーが指定されていない場合、またはそのフォルダーに有効な4D Volume Desktopが含まれない場合、ダブルクリックで起動可能なアプリケーションをビルドするボタンは無効になっています。

4D Volume Desktopファイルを含むフォルダーを選択します:

- Windowsでは4D Volume Desktop.4DEや4D Volume Desktop.RSR、その他動作に必要なファイルやフォルダーを含むフォルダーを選択します。
- Mac OSではソフトウェアパッケージとして4D Volume Desktopが提供されているので、このパッケージを選択します。

4D Volume Desktopフォルダーを選択するには**[...]**ボタンをクリックして表示されるフォルダーを選択ダイアログで、4D Volume Desktopフォルダー(Windows) やパッケージ (Mac OS) を選択します。
フォルダーが選択されるとその完全パス名が表示され、そこに4D Volume Desktopが含まれていればビルドボタンが有効になります:




### 生成されるファイル

実行可能アプリケーションをビルドするには**ビルド**をクリックします。

4Dは自動で"保存先"フォルダーに**Final Aplication**フォルダーを作成し、その中にアプリケーション名サブフォルダーを作成します。

"アプリケーション名"エリアに"MyAppli"と指定した場合、このフォルダー内には以下の要素がコピーされます:

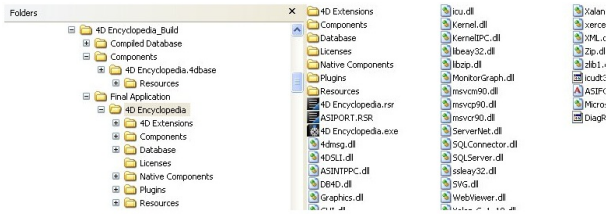
- Windows
  - MyAppli.exe: 実行可能ファイル、そしてMyAppli.RSR: アプリケーションリソースファイル。
  - 4D ExtensionsおよびResourcesフォルダー、その他さまざまなライブラリ (DLL) とアプリケーション実行に必要なファイル。
  - Databaseフォルダー: この中にはコンパイル済みストラクチャーを構成するDatabaseName.4DCやDatabaseName.RSRファイル、そしてデータベースのResourcesフォルダーがコピーされます。
  - (オプション) データベースに含まれるコンポーネントやプラグインが配置されたComponentsフォルダーおよびPlugInsフォルダー。この点に関する詳細はプラグイン & コンポーネントページを参照してください。
  - アプリケーションに統合されたXML形式のライセンスファイルが含まれるLicensesフォルダー。この点に関する詳細はライセンスページを参照してください。
  - 4D Volume Desktopフォルダーに追加されたその他の項目 (4D Volume Desktopフォルダーのカスタマイズ参照)。

  これらすべての項目が同じフォルダー内にコピーされます。

- Mac OS
  - MyAppli.appという名称のソフトウェアパッケージに、プラグインやコンポーネント、ライセンスなど必要な項目がすべてコピーされます。プラグインやコンポーネントの統合に関する詳細はプラグイン & コンポーネントページを参照してください。ライセンスの統合に関してはライセンスページを参照してください。.

  **注:** Mac OSでは、Application fileコマンドはソフトウェアパッケージ内の"Contents:Mac OS"フォルダー内にコピーされるApplicationNameファイルのパス名を返します。ソフトウェアパッケージの"Contents:Resources"フォルダー内の.compファイルではありません。

ダブルクリック起動可能なアプリケーションの典型的な構造は以下の通りです (Windows):




### 4D Volume Desktopフォルダーのカスタマイズ

ダブルクリックで起動可能なアプリケーションをビルドする際、4Dは4D Volume Desktopフォルダーの内容を**Final Application**内のアプリケーション名サブフォルダーにコピーします。必要に応じて元の4D Volume Desktopフォルダーの内容をカスタマイズできます。例えば:

- 特定の言語バージョンに対応する4D Volume Desktopをインストールする;
- PlugInsフォルダーにカスタムプラグインを配置する;

**注:** Mac OSで4D Volume Desktopはソフトウェアパッケージ形式で提供されます。内容を変更するにはパッケージを開かなくてはなりません (アイコン上を**Control+click**)。

### Webファイルの場所

ダブルクリックで起動可能なアプリケーションをWebサーバーとして使用する場合、Webフォルダーやファイルは特定の場所にインストールする必要があります:

- cert.pemとkey.pemファイル (オプション): これらのファイルはSSL接続に使用されます。
- デフォルトWebルートフォルダー

Windows: Final Application*MyAppli*\Database
Mac OS: MyAppli.appソフトウェアパッケージと同階層

## クライアント/サーバーページ

4Dを使用して、クライアントの自動更新をサポートするクロスプラットフォームなクライアント/サーバーアプリケーションをビルドできます。この設定はアプリケーションビルドダイアログの**クライアント/サーバー**ページで行います。

### クライアント/サーバーアプリケーションとは

クライアント/サーバーアプリケーションとはダブルクリックで起動可能なサーバーアプリケーションと、そのサーバーに接続可能な専用クライアントアプリケーションの組み合わせを意味します。

クライアント/サーバーアプリケーションをビルドするには以下の3つの項目が必要です:

- コンパイルされた4Dストラクチャー
- 4D Serverアプリケーション
- 4D Volume Desktopアプリケーション (Mac OS / Windows)

ビルドを行うと、クライアント/サーバーアプリケーションは2つのカスタマイズされたパーツ (サーバーと、各クライアントにインストールするクライアント) で構成されます。比較のために記載すると、ビルドされていない標準の展開時には4D Serverアプリケーション、ストラクチャーファイル、データファイル、リモートモードとして使用する4Dアプリケーションが必要です。

クライアント/サーバーアプリケーションは起動や接続処理が簡易になるようカスタマイズされています:

- サーバーを起動するには、サーバーアプリケーションをダブルクリックします。ストラクチャーを選択する必要はありません。
- クライアントを起動するには、クライアントアプリケーションをダブルクリックします。するとサーバーアプリケーションに直接接続します。接続ダイアログで サーバーを選択する必要はありません。接続に失敗した場合、ユーザーは標準の接続ダイアログでサーバーアドレスを入力できます。
  クライアントアプリケーション起動時に**Option** (Mac OS) や **Alt** (Windows) キーを押しておくと、標準接続ダイアログを強制的に表示させられます。
  サーバーに対応するクライアントのみが接続を行えます。標準の4Dアプリケーションを使用してサーバーに接続を試みると、接続は拒否されエラーが返されます。

さらにクライアント/サーバーアプリケーションではクライアント側をネットワーク越しに自動で更新できるよう設定を行うことができます。この点についてはサーバーアプリケーション内部のクライアントアプリケーションをコピーを参照してください。

また、ランゲージコマンドのシーケンスを使用することによって、サーバーパートのアップデートを自動化することも可能です(サーバーまたはシングルユーザー向けアプリの自動アップデート を参照して下さい)。

### クライアント/サーバーアプリケーション

- **サーバーアプリケーションをビルド**: ビルドフェーズでサーバー部分をビルドするためにはこのオプションを選択します。使用する4D Serverアプリケーションの場所を選択しなければなりません。この4D Serverはビルドを行うプラットフォームに対応していなければなりません (つまり、例えばWindows用のサーバーをビルドするためにはWindows上でビルドを実行する必要があります)。
  4D Serverフォルダーを選択するには**[...]**ボタンをクリックします。Mac OSでは4D Serverパッケージを選択します。

- **現在のバージョン**: 生成されるアプリケーションのバージョン番号を指定するために使用されます。このバージョン番号をもとに、クライアントアプリケーションからの接続を受け入れたり拒否したりできるようになります。クライアントとサーバーアプリケーションで互換性のある番号の範囲はXMLで設定します (参照)。

- **クライアントアプリケーションをビルド**: ビルドフェーズでクライアント部分をビルドするためにはこのオプションを選択します。使用する 4D Volume Desktopアプリケーションの場所を選択しなければなりません。この 4D Volume Desktopはビルドを行うプラットフォームに対応していなければなりません。他方のプラットフォームのクライアントアプリケーションをビルドするためには、同じ処理をそのプラットフォーム上でも行わなければなりません。この点については後述します。
  4D Volume Desktopフォルダーを選択するには**[...]**ボタンをクリックします。Mac OSでは4D Volume Desktopパッケージを選択します。

#### 4D Serverや4Dクライアントフォルダーのカスタマイズ

実行可能クライアント/サーバーアプリケーションビルド中に、4Dは4D Serverフォルダーの内容をServerサブフォルダーに、4D Volume Desktopフォルダーの内容をClientサブフォルダーにコピーします。元の4D Serverと4D Volume Desktopの内容を必要に応じてカスタマイズできます。

例えば以下のようなことが可能です:

- 特定の言語に対応した4D Serverをインストールする
- PlugInsフォルダーにプラグインを追加する

**注**: Mac OSでは4D Serverがパッケージとして提供されます。内容を見るためにはまず (**Control+click**してメニューから選択し) パッケージを開かなくてはなりません。

### サーバーアプリケーション内部のクライアントアプリケーションをコピー

このエリアのオプションを使用して、新しいバージョンのアプリケーションがビルドされた際の、ネットワーク越しのクライアント/サーバーアプリケーションのクライアント部分更新メカニズムを設定できます。

- **Windows/Macintoshクライアントアプリケーションの自動更新を許可する:** これらのオプションを選択すると、ネットワーク越しの自動クライアント更新を有効にすることができます。
  クロスプラットフォームなクライアントアプリケーションを作成するには、ビルドを行うのとは別のプラットフォーム用の4D Volume Desktopアプリケーションを選択する必要があります。例えばWindows上で**[...]**ボタンをクリックし、Mac OS用の4D Volume Desktop.appフォルダーを選択します。

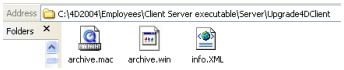
クライアントアプリケーションの更新は、新しいバージョンの4D Serverを使用する場合になどに必要となります。

ストラクチャーの更新は常に自動で各クライアントマシンに適用される点に留意してください。

自動更新機能を使用すれば、手作業で各クライアントマシンに新バージョンの4Dをインストールする手間を省くことができます。

#### 更新通知の表示

この設定を有効にしたサーバーが更新されると、各クライアントマシン上に自動で通知が行われます。これは以下のように動作します: アプリケーションビルダーを使用して新しいバージョンのクライアント/サーバーアプリケーションをビルドする際、新しいクライアントは***ApplicationName* Server**内の**Upgrade4DClient**フォルダーに圧縮して格納されます (Mac OSでは、これらのフォルダーがサーバーパッケージ内に配置されます)。クロスプラットフォームクライアントアプリケーションを生成する処理を行った場合、各プラットフォーム用に更新ファイルが含められます:




クライアントアプリケーション更新通知を行うために、古いサーバーアプリケーションを新しいバージョンで置き換えて起動します。残りの処理は自動で行われます。

クライアント側では古いバージョンのクライアントが更新されたサーバーに接続を試みると、新しいバージョンが利用可能な旨を表示するダイアログがクライアントマシン上に表示されます。ユーザーはダイアログを受け入れるかキャンセルできます。

- ユーザーが**OK**をクリックすると、新バージョンがネットワーク越しにクライアントマシンにダウンロードされます。ダウンロードが完了すると古いクライアントアプリケーションが閉じられて、新しいバージョンが起動しサーバーに接続します。古いバージョンはゴミ箱に移動されます。
- ユーザーが**キャンセル**をクリックすると、更新はキャンセルされます。古いバージョンのクライアントの接続がプロジェクトXMLで許可された番号の範囲外である場合 (後述)、アプリケーションは閉じられて、接続を行うことはできません。そうでなければ接続が行われます。

#### 自動更新を強制する

更新のダウンロードをキャンセルさせたくない場合、例えば新しいメジャーバージョンの4D Serverを使用するような場合 (この場合古い4Dが4D Serverに接続するこてゃ不可能です)、新しいバージョンのクライアントアプリケーションを各クライアントマシンに必ずインストールしなければなりません。

更新を強制するためには、サーバーアプリケーションと互換性のあるバージョン番号の範囲からクライアントアプリケーションの現在のバージョン番号を除外します。このようにすれば更新メカニズムは更新されていないクライアントからの接続を拒否します。例えばクライアントサーバーアプリケーションの新しいバージョン番号が6の場合、バージョン番号が5以下のクライアントアプリケーションを許可しないようにできます。

現在のバージョン番号はアプリケーションビルドダイアログのクライアント/サーバーページで設定できます (前述)。接続を許可するバージョン番号の範囲はプロジェクトXMLファイルで指定します。この点に関する詳細はを参照してください。

#### エラーが発生する場合

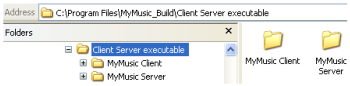
クライアントアプリケーションの更新を実行できなかった場合、クライアントマシンには以下のメッセージが表示されます: "クライアントアプリケーションの更新に失敗しました。アプリケーションは終了します。"

このエラーが発生する原因は複数ありえます。このエラーが表示されるような場合はまずいかの点をチェックしてみてください:

- **パス名**: アプリケーションビルドダイアログやXMLキー (例えば *ClientMacFolderToWin*) で指定されたパス名の有効性をチェックしてください。特に4D Volume Desktopへのパスをチェックしてください。
- **読み書き権限**: クライアントマシン上でカレントユーザーがクライアントアプリケーションを更新する書き込みアクセス権を持っているか確認してください。

### 生成されるファイル

クライアント/サーバーアプリケーションをビルドすると、保存先フォルダー内に**Client Server executable**という名前の新しいフォルダーが作成されているはずです。このフォルダーにはさらに2つのサブフォルダー、ApplicationName ClientとApplicationName Serverがあります:



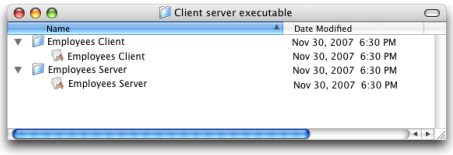

**注:** エラーが発生した場合これらのフォルダーは作成されません。そのような場合はエラーの原因を特定するためにログファイル (ログファイル参照)の内容を調べてください。

**ApplicationName Client** フォルダーにはアプリケーションビルダーが実行されたプラットフォームに対応するクライアントアプリケーションがコピーされます。このフォルダーを各クライアントにインストールします。**ApplicationName Server**フォルダーにはサーバーアプリケーションがコピーされます。

これらのフォルダーの内容はカレントのプラットフォームにより異なります:

- Windowsでは各フォルダーに*ApplicationName Client.exe* (クライアント用) あるいは*ApplicationName Server.exe* (サーバー用) という名前の実行可能ファイル、およびそれぞれに対応する.rsrファイルが作成されます。さらにこれらのフォルダーにはアプリケーション実行のために必要な様々なファイルやフォルダー、および元の4D Serverや4D Volume Desktopに追加されたカスタマイズ項目がコピーされます。

- Mac OSでは各フォルダーに*ApplicationName Client* (クライアント用) と*ApplicationName Server* (サーバー用) という名前のアプリケーションパッケージが作成されます。各パッケージには動作に必要なすべてのファイルが含まれます。Mac OSではアプリケーションを実行するためにパッケージをダブルクリックします。




**注:** ビルドされたMac OSパッケージの内容を表示するにはアイコンを**Control+クリック**して、"パッケージの内容を表示"を選択します。

"クライアントの自動更新を許可する"オプションを選択している場合、***ApplicationName* Server**フォルダー/パッケージには追加で**Upgrade4DClient**サブフォルダーが作成されます。このフォルダーにはクライアントアプリケーションが圧縮されて格納されます。このファイルは自動クライアントアプリケーション更新時に使用されます。

#### Webファイルの場所

サーバーやクライアントをWebサーバーとして使用する場合、Webサーバーが使用するファイルを特定の場所に配置しなければなりません:

- cert.pem と key.pem ファイル (オプション): これらのファイルはSSL接続で使用されます。
- デフォルトWebルートフォルダー (WebFolder)

  **Windows**

  - サーバーアプリケーション: これらの項目はサーバーアプリケーションフォルダー内にインストールします: Client Server executable*ApplicationName* Server\Server Database サブフォルダー
  - クライアントアプリケーション: これらの項目はクライアントアプリケーションフォルダー内にインストールします: Client Server executable*ApplicationName* Client サブフォルダー

  **Mac OS**

  - サーバーアプリケーション: これらの項目は*ApplicationName* Serverソフトウェアパッケージと同階層にインストールします。
  - クライアントアプリケーション: これらの項目は*ApplicationName* Clientソフトウェアパッケージと同階層にインストールします。

## プラグイン & コンポーネントページ

**プラグイン & コンポーネント** ページではビルドするアプリケーションで使用するプラグインやコンポーネントを設定できます。4Dプラグインやコンポーネントについてはプラグインやコンポーネントのインストールを参照してください。

このページには現在4Dにロードされているプラグインやコンポーネントがリストされます:

| アクティブ | プラグイン & コンポーネント | ID | タイプ |
|---|---|---|---|
| ☑ | 4D for OCI | 11900 | Plugin |
| ☑ | 4D Internet Commands | 15010 | Plugin |
| ☑ | 4D ODBC Pro | 13500 | Plugin |
| ☑ | 4D_Pack | 11999 | Plugin |
| ☑ | 4D View | 13000 | Plugin |
| ☑ | 4D Write | 12000 | Plugin |
| ☑ | 4D Progress | 0 | Component |
| ☑ | 4D SVG | 0 | Component |
| ☑ | 4D Widgets | 0 | Component |

**タイプ** 列には、その要素がプラグインであるかコンポーネントであるかが表示されます。

**アクティブ** 列では、その行の項目をビルドされるアプリケーションに統合するかどうかを指定できます。デフォルトですべての項目が選択されています。プラグインやコンポーネントをアプリケーションから除外するには、チェックボックスの選択を外します。

実行可能アプリケーションにその他のプラグインやコンポーネントを統合したい場合、4D Serverや4D Volume Desktopの**Plugins** や **Components** フォルダーにそれらを配置します。ソースアプリケーションフォルダーの内容をコピーするメカニズム (4D Volume Desktopフォルダーのカスタマイズ参照) により、どんなタイプのファイルでもアプリケーションに統合できます。

- 同じプラグインの異なるバージョンが見つかった場合 (4Dレベルおよびソースアプリケーションフォルダーレベル両方にプラグインがインストールされていた場合)、4D Volume Desktop/4D Serverフォルダーにインストールされているバージョンが優先されます。
- 他方同じコンポーネントが両方にインストールされていた場合、アプリケーションを開くことはできません。

**注:** 配布するアプリケーションでプラグインやコンポーネントを使用するためには、それぞれ適切なライセンスが必要です。

## ライセンスページ

**ライセンス&証明書**のページでは、以下の様なことができます:

- シングルユーザー向けスタンドアロンアプリケーションへと統合したいライセンス番号を指定することができます。
- OS X環境下では、証明書を使用してアプリケーションに署名をすることができます。

### Licences

このページではアプリケーションに統合するのに使用できる配付ライセンスの一覧を表示します。

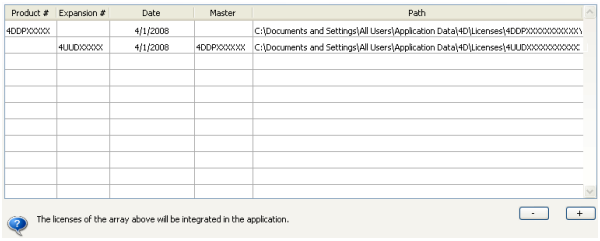
デフォルトでリストは空です。有効なアプリケーションをビルドするためには*4D Developer Professional*ライセンスと、その開発ライセンスに対応する*4D Desktop Volume*ライセンスを指定しなければなりません。

ライセンスを追加または取り除くにはウィンドウ下部の[+]または[-]ボタンをクリックします。

[+] ボタンをクリックするとファイルを開くダイアログが表示され、マシンのLicensesフォルダーの内容が表示されます。このフォルダーの場所については Get 4D folderコマンドの説明Licenses Folderを参照してください。

開発ライセンスとそれに対応した配布ライセンスを選択しなくてはなりません。これらのファイルは*4D Developer Professional*ライセンスや*4D Desktop Volume*ライセンスをアクティベーションした際この場所にコピーされます。

ファイルを選択すると、リストに選択内容が反映されます (ライセンス番号、有効期限等):




ライセンスが有効でない場合、警告が表示されます。

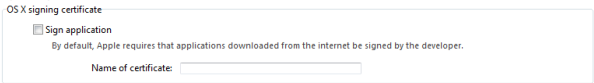
必要なだけ有効なファイルを選択することができます。実行可能アプリケーションをビルドする際に4Dは最も適切なライセンスを使用します。

アプリケーションビルド後、配布ライセンスファイルが実行可能ファイルと同階層(Windows)やパッケージ内(Mac OS)に自動でコピーされます。

### Certification of applications under OS X

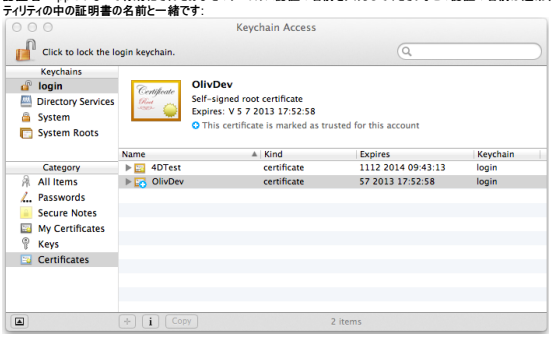
アプリケーションビルダーは、OS X 環境下において組み込み4Dアプリに署名をする新機能を備えています(OS X 環境下のシングルユーザーアプリ、4D サーバーアプリおよびクライアントアプリ)。アプリケーションを署名する事により、OS X において「Mac App Store と確認済みの開発元からのアプリケーションを許可」のオプションが選択されているときに Gatekeeper の機能を使用してアプリケーションを実行することが可能になります(以下の"Gatekeeperについて"を参照して下さい)。

- **アプリケーションに署名**オプションにチェックをすると、OS X のアプリケーションビルド処理に認証が含まれます。4D はビルドの際に認証に必要な要素の有無をチェックします:

  このオプションはWindowsとOS X両方の環境で表示されますが、OS Xの場合においてのみ有効です。

- **認証名**: Apple によって有効化されたあなたのデベロッパ認証の名前を入力してください。この認証の名前は通常、キーチェーンアクセスユーティリティの中の証明書の名前と一緒です:

Apple からデベロッパ認証を取得するためには、**キーチェーンアクセス**のメニューのコマンドを使用するか、以下のリンクへ移動して下さい:

http://developer.apple.com/library/mac/#documentation/Security/Conceptual/CodeSigningGuide/Procedures/Procedures.html

**注:** この証明書の取得には Apple の codesign ユーティリティが必要になります。このユーティリティはデフォルトで提供されており、通常"/usr/bin/codesign" フォルダの中にあります。エラーが起きた際には、このユーティリティがディスク上にあるかどうかを確認して下さい。

#### Gatekeeperについて

*Gatekeeper*とは OS X のセキュリティ機能で、インターネットからダウンロードしてきたアプリケーションの実行を管理するものです。OS X の10.8 Mountain Lion 以降、「**Mac App Store と確認済みの開発元からのアプリケーションを許可**」のオプションがデフォルトで選択されています(Apple は最低レベルの「すべてのアプリケーションを許可」のオプションを選択することを推奨していません )。もしダウンロードしてきたアプリケーションが Apple Store からダウンロードしたものではない、または署名されていない場合には実行が拒否されます。

4D アプリケーションビルダーの**アプリケーションに署名**の新機能は、このオプションとデフォルトで互換性のあるアプリケーションを生成可能にします。

## サーバーまたはシングルユーザー向けアプリの自動アップデート

旧バージョンの4Dでは、サーバーアプリとシングルユーザー用組み込みアプリは、ユーザーの操作(またはカスタムのシステムルーチンのプログラミング)が必須でした。組み込みアプリの新しいバージョンができる度に製品のアプリを終了し、古いファイルを手動で新しいファイルと置き換え、アプリケーションを再起動してカレントのデータファイルを選択しなければなりませんでした。

二つの新しいランゲージコマンドを使用して、これらの処理を自動化することができます。 SET UPDATE FOLDERとRESTART 4Dです。また、モニターオペレーションには Get last update log path を使用することができます。これらのコマンドは以下のような自動アップデートシークエンス機能をアプリケーションに実装することを目的としています。FTP サーバーに新しいファイルの有無を定期的に確認するメニューコマンドまたはバックグラウンドで実行されるプロセスとして実装することができます。

新コマンドを使用したアップデートの手順は以下の通りです:

1. FTPサーバーなどを通じて、サーバーアプリケーションまたはシングルユーザー向け組み込みアプリケーションの新バージョンを稼働中のマシンへと転送します。
2. 稼働中のアプリの中で新コマンド SET UPDATE FOLDER を呼び出します。このコマンドはカレントアプリケーションのアップデート"予定"のバージョンを置くフォルダの場所を指定します。
   また、このフォルダの中に稼働中のバージョンのカスタムエレメンツ(ユーザーファイル)を再コピーすることも可能です。
3. 稼働中のアプリケーション内で RESTART 4D コマンドを呼び出します。このコマンドは"アップデーター"というユーティリティープログラムの実行をトリガーします。このプログラムはカレントアプリケーションを終了し、指定があった場合にはカレントアプリケーションをアップデート"予定"のバージョンと置き換え、カレントのデータファイルでアプリケーションを再起動します。古いバージョンの方は改名されます。

**注:**

- この機能はWindowsのサーバーアプリケーションとも互換性があります(データベースをサービスとして登録を参照して下さい)。
- また、Windows環境下で保護されたファイルが使用できるようにインストール権限を引き上げるXMLキーもあります(詳細はアプリケーションビルド設定ファイル マニュアルを参照して下さい )。

**アップデートログ**

インストール処理は、ターゲットマシン上の組み込みアプリ(クライアントアプリまたはシングルユーザー向けアプリ)のアップデートオペレーションの詳細を記したログファイルを生成します。このファイルはインストール処理中にエラーが起きたときにそれを解析するために有用です。

アップデートログのファイル名は **YYYY-MM-DD_HH-MM-SS_log_<sequence>.txt** となっています。例えば、2013年8月25日の14:23に作成されたログのファイル名は **2013-08-25_14-23-00_log_1.txt** となります。

このファイルは"Updater"アプリケーションフォルダの中に作成されます。具体的には以下の場所に作成されます:

- OS Xの場合:
  {*userName*}/Library/Appplication Support/{*ProductName*}/4D/Updater/

- Windowsの場合:
  \{*userName*}\AppData\Roaming\{*ProductName*}\4D\Updater\

上記の場所は Get last update log path コマンドを使用することによっていつでも確認することができます。

## スタンドアロンアプリケーションアイコンのカスタマイズ

4DはFinder (Mac OS) やエクスプローラー (Windows) に表示されるアイコンとして、実行可能アプリケーションにデフォルトアイコンを割り当てますが、各アプリケーションごとにこのアイコンをカスタマイズできます。

**Mac OSの場合**

icnsタイプのアイコンファイルを作成し、それをインタープリターストラクチャーファイルと同階層に配置します。

**注：** Apple, Inc.よりicnsアイコンファイルを作成するツールが提供されています。

アイコンファイルの名前はストラクチャーファイル名+".icns"拡張子でなければなりません。

4Dは自動でこのファイルを認識し、アイコンとして使用します (.icnsファイルを*ApplicationName*.icnsに名称変更してResourcesフォルダーにコピー、さらにinfo.plistファイルの*CFBundleFileIcon*エントリを更新します)。

**Windowsの場合**

.ico拡張子のアイコンファイルを作成し、それをインタープリターストラクチャーファイルと同階層に配置します。

アイコンファイルの名前はストラクチャーファイル名+".ico"拡張子でなければなりません。

4Dは自動でこのファイルを認識し、アイコンとして使用します。

# 4Dコンポーネントの開発とインストール


概要

コンポーネントのインストールと互換性

コンポーネントの開発

コンポーネントとホストデータベースの相互作用

## 概要

4D のコンポーネントは、異なるデータベースにインストール可能な、1つ以上の機能を持つ4D オブジェクトの集合です。例えばメールの送信や受信を行い、それらを4Dデータベースに格納するための機能を持ったコンポーネントを作成できます。

4D コンポーネントの作成とインストールは直接4Dを使用して行われます。4Dでは、コンポーネントはプラグインのように扱われ、以下の原則が適用されます:

- コンポーネントは、標準のアーキテクチャーまたはパッケージの形をした通常のストラクチャーファイル ( コンパイルまたは非コンパイル) で構成されます (.4dbase拡張子参照)。
- データベースにコンポーネントをインストールするには、データベースのストラクチャーファイルと同階層に置かれる"Components" フォルダーにコンポーネントをコピーします。ショートカット (Windows) やエイリアス(Mac OS) を使用することもできます。アンインストールするには、コンポーネントをフォルダーから取り除きます。
- コンポーネントで標準のテーブルやデータファイルを使用することはできません。しかし外部データベースのメカニズムを使用すればテーブルやフィールドを作成し、そこにデータを格納したり読み出したりすることができます。外部データベースはメインの4Dデータベースとは独立して存在し、SQLコマンドでアクセスします。詳細はエクスターナルデータベースについてを参照してください。

コンポーネントを作成するには適切なライセンスが必要です。他方コンポーネントを使用するためにインストールするにはライセンス上の制限はありません。

### 定義

4Dにおける新しいコンポーネント管理メカニズムでは、以下の用語とコンセプトを知っている必要があります:

- **マトリクスデータベース:** コンポーネント開発に使用する4D データベース。マトリクスデータベースは特別な属性を持たない標準のデータベースです。マトリクスデータベースはひとつのコンポーネントを構成します。
  マトリクスデータベースは、コンポーネントを使用するデータベース ( ホストデータベース) のComponents フォルダーにコピーされます。コンパイルされていてもいなくてもかまいません。

- **ホストデータベース:** コンポーネントがインストールされ、それを使用するデータベース。

- **コンポーネント:** ホストデータベースのComponentsフォルダーにコピーされ、その内容がホストデータベースで使用されるマトリクスデータベース ( コンパイル済みまたは非コンパイル)。

- **プロジェクトフォーム**: テーブルにリンクされていないフォーム。プロジェクトフォームは特にコンポーネントの生成に使用できます。プロジェクトフォームに関する詳細はテーブルフォームとプロジェクトフォームを参照してください。

- **テーブルフォーム** ("標準"フォーム): テーブルに属するフォーム。このタイプのフォームをコンポーネントで使用することはできません。

データベースは" マトリクス" にも" ホスト" にもなりえます。言い換えれば、マトリクスデータベースが1 つ以上のコンポーネントを使用できます。しかしコンポーネントが" サブコンポーネント" を使用することはできません。

### コンポーネントの保護: コンパイル

コンポーネントとしてインストールされたマトリクスデータベースのプロジェクトメソッドは、デフォルトでホストデータベースからアクセスが可能です。特に:

- 共有プロジェクトメソッドがエクスプローラーのメソッドページに存在し、かつ ホストデータベースのメソッドから呼び出し可能な場合 (プロジェクト メソッドの共有参照)。エクスプローラーのプレビューエリアでそれらの内容を選択してコピーすることが可能です。またデバッガーでも内容を見ることができます。しかしそれらをメソッドエディター上で開いたり編集したりすることはできません。
- マトリクスデータベースの他のプロジェクトメソッドはエクスプローラーに現れません。しかしホストデータベースのデバッガーには内容が表示されます。

コンポーネントのプロジェクトメソッドを効果的に保護するには、マトリクスデータベースを**コンパイル**して、インタプリターコードを含まない .4dc ファイルとして提供します。コンパイルされたマトリクスデータベースがコンポーネントとしてインストールされると:

- 共有プロジェクトメソッドはエクスプローラーのメソッドページに表示され、ホストデータベースのメソッドから呼び出すことができます。しかしその内容はプレビューエリアにもデバッガーにも表示されません。
- マトリクスデータベースの他のプロジェクトメソッドは全く表示されません。

## コンポーネントのインストールと互換性

### Components フォルダー

4Dデータベースにコンポーネントをインストールするには、ホストデータベースの**Components**フォルダーにマトリクスデータベースのストラクチャーをコピーします。**Components** フォルダーは以下二つの場所に配置できます:

- 4D実行アプリケーションレベル: この場合、そのコンポーネントはそのアプリケーションで開かれたすべてのデータベースで利用できます。
- データベースストラクチャーファイルレベル: この場合、コンポーネントはそのデータベースでのみ利用できます。

これに関する詳細はPlugInsおよびComponentsフォルダーの場所を参照してください。

4D は**Components**フォルダー内で、.4db ( インタープリターマトリクスデータベース)、.4dc ( コンパイル済みマトリクスデータベース) または .4dbase( パッケージタイプのマトリクスデータベース) マトリクスデータベースを探します。データファイルやユーザーストラクチャファイル (.4DA) などは無視されます。これらのマトリクスデータベースのショートカットやエイリアスを使用できます。これはコンポーネントの開発時に便利です。マトリクスデータベースに対する更新はすぐにホストデータベースで有効になります。

**Components** フォルダーには、(xliff, ピクチャーなど) コンポーネントの動作に必要なカスタムファイルやフォルダーを置くことができます。他方、プラグインやComponent サブフォルダーを置くことはできません。これらが置かれていた場合、4D はそれらを無視します。
コンポーネントで使用されるプラグインは、ホストデータベースかまたは4D にインストールします。

### インタープリター / コンパイル / Unicode

インタープリターモードで動作するホストデータベースは、インタープリターまたはコンパイル済みどちらのコンポーネントも、またUnicode モードであっても無くても、使用できます。一つのホストデータベースに、インタープリターとコンパイル済み両方のコンポーネントをインストールできます。しかし、複数のコンパイル済みコンポーネントが存在する場合、それらは同じUnicode のモードで実行されなければなりません。

コンパイルモードで実行されるホストデータベースでは、インタープリターのコンポーネントを使用できません。この場合、コンパイル済みコンポーネントのみが利用可能です。またUnicode のモードはホストデータベースとコンポーネントで同じでなければなりません。

以下の表でこの点についてまとめます:

| | | インタプリターコンポーネント | | コンパイル済みコンポーネント | |
|---|---|---|---|---|---|
| | | Unicode | 非Unicode | Unicode | 非Unicode |
| **インタープリターホストデータベース** | Unicode | X | X | X (*) | X (*) |
| | 非Unicode | X | X | X (*) | X (*) |
| **コンパイル済みホストデータベース** | Unicode | - | - | X | - |
| | 非Unicode | - | - | - | X |

(*) 複数のコンパイル済みコンポーネントがインストールされる場合、それらは同じUnicode モードで動作しなければなりません。

**注:**

- インタープリターコンポーネントがインストールされたインタープリターホストデータベースは、それがインタープリターコンポーネントのメソッドを呼び出さなければコンパイルできます。そうでない場合、コンパイル... メニューを選択すると、警告ダイアログが表示され、コンパイルはできません。
- インタープリターメソッドはコンパイル済みメソッドを呼び出せますが、逆はできません。これを行うためにはEXECUTE METHODやEXECUTE FORMULAコマンドを使用します。

コンポーネント間及びホストデータベースとコンポーネント間の相互作用についてはコンポーネントとホストデータベースの相互作用を参照してください。

### Mac OS / Windows

Mac OS で開発されたインタープリターコンポーネントをWindows 環境にインストールすること、及びその逆が可能です。他方、コンパイルされたコンポーネントは、コンパイルされたのと同じプラットフォームでのみ利用可能です。両プラットフォームで利用可能
するためには、両プラットフォーム用にコンパイルします。

### クライアント/サーバー

サーバーデータベースにインストールされたコンポーネントは、プラグインと同様のメカニズムを使用して、自動でクライアントマシンに転送されます。

クライアント/ サーバーモードで、クライアント側のコンポーネントのコピーを更新することはお勧めできません。変更はローカルに保存され、サーバーマシンのコンポーネントは更新されません。

### スタートアップ時の コンポーネントのロード

コンポーネントはホストデータベースが開かれるときにロードされます。

- コンポーネントが一致しないコンパイル済みコードとインタープリターコードを含んでいる場合、エラーメッセージが表示され、そのコンポーネントはホストデータベースにロードされません。
- 起動時にコンポーネントが見つからない場合、ホストデータベースは開かれずエラー -10509 (データベースを開けません) が生成されます。データベース起動時にCOMPONENT LISTコマンドを使用してコンポーネントの存在をチェックできます。特定のコンポーネントの有無にかかわらず実行可能なホストデータベースを開発したい場合、以下のコードを使用します:

```
ARRAY TEXT($arrComponents_Txt;0)
COMPONENT LIST($arrComponents_Txt)
If(Find in array($arrComponents_Txt;"ComponentA")>0) // コンポーネントAは存在しなくてもよい
   EXECUTE METHOD("ComponentAMethod")
End if
```

### 初期化のコードの実行

コンポーネントは、ホストデータベースを開いたときまたは閉じたときに、自動的に4Dコードを実行することができます。これによって例えば、ホストデータベースに関連する設定やユーザーの状態などを読み出し・保存することができます。

初期化やデータベースを閉じるコードの実行は、On Host Database Event データベースメソッド を使用して行われます。詳細な情報に関しては、4D *Language Reference* マニュアル内のこのコマンドの詳細を参照して下さい。

セキュリティ上の理由から、このデータベースメソッドを使用可能にするためには、その実行をホストデータベースで明示的に許可する必要があります。詳細に関しては、35P "4D components" を参照して下さい。

セキュリティ上の理由から、このデータベースメソッドを実行するためには、ホストデータベースにおいてその実行が明示的に認証されなければなりません。このためには、データベース設定画面の セキュリティページ ページ内の、"**コンポーネントの "On Hose Database Event" メソッドを実行**"のオプションにチェックを入れます:




### 名前の衝突 ( メソッドのマスク)

他の共有オブジェクトと異なり (共有及び非共有オブジェクト参照)、共有プロジェクトメソッドはデータベースに" 物理的に"存在します。つまりコードの実行時に作成されるわけではありません。

ゆえに、コンポーネントの共有プロジェクトメソッドと、ホストデータベースのプロジェクトメソッドが同じ名前を持つと、名前の衝突が発生します。この場合、ホストデータベースのコンテキストでコードが実行されると、ホストデータベースのメソッドが呼び出されます。これはつまり、コンポーネントメソッドをカスタムメソッドで"マスク" して、例えば異なる機能を実装することが可能であることを示しています。
もちろん、コンポーネントの中でコードが実行されている場合は、コンポーネントのメソッドが呼び出されます。

このマスキングは、ホストデータベースのコンパイル時に警告によって通知されます。

**注:** 2つのコンポーネント間で共有メソッド名に衝突があった場合、ホストデータベースをコンパイルする際にエラーが生成されます。

## コンポーネントの開発

コンポーネントは4D データベースの形で構成されるので、コンポーネントの開発はデータベースへの機能実装と同じです。しかしながら、コンポーネントの持つ性質のため、制限や特別なルールがあります。

### 利用可能又は利用不可能なオブジェクト

コンポーネントはプロジェクトメソッド, プロジェクトフォーム, メニューバー, 選択リスト, ライブラリピクチャーなど、4D のほとんどのオブジェクトを呼ぶことができます。

以下のオブジェクトのみ、コンポーネントで使用することはできません:

- 標準のテーブルとフィールド (しかし外部データベースのテーブルやフィールドを作成し利用することはできます)
- テーブルフォームとそのフォームメソッド ( 他方コンポーネントはホストデータベースのテーブルフォームを呼び出すことはできます)
- ユーザーフォーム
- データベースメソッドとトリガー

マトリクスデータベースにこれらの要素が含まれている場合でも、それらを削除する必要はありません。利用できないオブジェクトは単に無視されます。

**注:** マトリクスデータベースに設定されたユーザー＆グループやアクセス権は、ホストデータベースでは無視されます。

コンポーネント側で" 共有" されているプロジェクトメソッドのみがホストデータベースのデザインモードに表示され、選択することができます。他方、ホストデータベース側で" 共有" されているプロジェクトメソッドをコンポーネントから呼ぶことが可能です。詳細はプロジェクト メソッドの共有を参照してください。

コンポーネントから公開されたフォームは、ホストデータベースのサブフォームとして組み込むことが可能です。この点の詳細はフォームの共有を参照してください。

他のコンポーネントオブジェクト (公開されていないプロジェクトフォーム、選択リスト、メニュー等) はコンポーネント内で使用することはできますが、ホストデータベースからストラクチャーオブジェクトとしてアクセスすることはできません。特定のオブジェクトはホストデータベースとコンポーネント間で共有され、その他のオブジェクトは共有されません。詳細は共有及び非共有オブジェクトを参照してください。

コンポーネントは4D アプリケーション又はホストデータベースにインストールされたプラグインを使用することができます。コンポーネントフォルダにプラグインをインストールすることはできません。

マトリクスデータベースのデータベースメソッドや、一般設定 (Web フォルダ, 環境設定, その他) は一切参照されません。

### 使用できないコマンド

(読み込みのみで開かれるため) ストラクチャーファイルを更新する以下のコマンドをコンポーネントで使用することはできません。コンポーネント中で以下のコマンドを実行すると-10511, "CommandName コマンドをコンポーネントでコールすることはできません" のエラーが生成されます

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

**注:**

- Current form tableコマンドは、プロジェクトフォームのコンテキストで呼び出されるとNil を返します。ゆえにこのコマンドをコンポーネントで使用することはできません
- SQLデータ定義言語のコマンド (CREATE TABLE、DROP TABLE等) をコンポーネントのフレームワークで使用することはできません。

### フォームの使用

- 特定のテーブルに属さない" プロジェクトフォーム" のみが、コンポーネントで利用できます。マトリクスデータベースのすべてのプロジェクトフォームをコンポーネントで使用することができます。

- コンポーネントはホストデータベースのテーブルフォームを使用できます。この場合、コンポーネントのコードでフォームを指定する際に、テーブル名ではなく、テーブルへのポインターを使用しなければならないことに注意してください。
  **注:** コンポーネントはADD RECORDコマンドを使用すると、ホストデータベースのコンテキストで、ホストデータベースのカレントの入力フォームが表示されます。フォーム上に変数が配置されている場合、コンポーネントはその変数にアクセスできません (コンポーネントとホストデータベースの相互作用参照)。

- コンポーネントフォームをホストデータベースに対してサブフォームとして公開できます。これは特にグラフィックオブジェクトを提供するコンポーネントを開発できることを意味します。例えば4D社が提供するウィジェットはコンポーネント内でのサブフォームの利用に基づいています。この点についてはフォームの共有で説明しています。

### テーブルやフィールドの利用

コンポーネントはマトリクスデータベースのストラクチャーで定義されたテーブルやフィールドを使用することはできません。しかし**外部データベース**を作成し、そのテーブルやフィールドを必要に応じ利用することはできます。外部データベースの作成と管理はSQLを使用して行います。外部データベースは4Dデータベースで、主たる4Dデータベースからは独立していますが、主たる4Dデータベースから操作が可能です。外部データベースの利用は、一時的にそのデータベースをカレントデータベースにすることです。言い換えれば4Dが実行するSQLクエリのターゲットデータベースを切り替えます。外部データベースの作成はSQLのCREATE DATABASEコマンドを使用します。

外部データベースに関する詳細はエクスターナルデータベースについてを参照してください。

#### 例題

以下のコードはコンポーネントに実装されていて、外部データベースに対して3つの基本的なアクションを行います:

- 外部データベースを作成します (まだ存在しなければ)。
- 外部データベースにデータを追加します。
- 外部データベースからデータを読み込みます。

外部データベースの作成:

```
<>MyDatabase:=Get 4D folder+"\MyDB"
Begin SQL
        CREATE DATABASE IF NOT EXISTS DATAFILE :[<>MyDatabase];
        USE DATABASE DATAFILE :[<>MyDatabase];
        CREATE TABLE IF NOT EXISTS KEEPIT
        (
        ID INT32 PRIMARY KEY,
        kind VARCHAR,
        name VARCHAR,
        code TEXT,
        sort_order INT32
        );

        CREATE UNIQUE INDEX id_index ON KEEPIT (ID);

        USE DATABASE SQL_INTERNAL;

End SQL
```

外部データベースへのデータ書き込み:

```
$Ptr_1:=$2 // ホストデータベースへのデータアクセスはポインターを通じて行う
$Ptr_2:=$3
$Ptr_3:=$4
$Ptr_4:=$5
$Ptr_5:=$6
Begin SQL

        USE DATABASE DATAFILE :[<>MyDatabase];

        INSERT INTO KEEPIT
        (ID, kind, name, code, sort_order)
        VALUES
        (:[$Ptr_1], :[$Ptr_2], :[$Ptr_3], :[$Ptr_4], :[$Ptr_5]);
```

```
        USE DATABASE SQL_INTERNAL;

 End SQL
```

**外部データベースからデータを読み込み**:

```
 $Ptr_1:=$2 // ホストデータベースへのデータアクセスはポインターを通じて行う
 $Ptr_2:=$3
 $Ptr_3:=$4
 $Ptr_4:=$5
 $Ptr_5:=$6

 Begin SQL

    USE DATABASE DATAFILE :[<>MyDatabase];

    SELECT ALL ID, kind, name, code, sort_order
    FROM KEEPIT
    INTO :$Ptr_1, :$Ptr_2, :$Ptr_3, :$Ptr_4, :$Ptr_5;

    USE DATABASE SQL_INTERNAL;

 End SQL
```

**リソースの使用**

コンポーネントはリソースを使用することができます (" 伝統的な"MacOS リソースやXLIFF タイプのファイル)。

新しいリソース管理の原則に従い (データベースアーキテクチャー参照)、コンポーネントのリソースファイルは、コンポーネントの.4db や.4dc ファイルと同階層の**Resources** フォルダーに置かれなければなりません。コンポーネントが.4dbase 形式の場合 ( 推奨されるアーキテクチャー)、Resources フォルダは .4dbase フォルダーの中に置かれます。

自動メカニズムが有効となり、コンポーネントのResources フォルダー内で見つかったXLIFF ファイルは、このコンポーネントによってロードされます。( 互換性のため) コンポーネントは、.4db や.4dc と同階層にある.rsr ファイルに格納された" 伝統的な" Mac OS リソースも自動的に使用します。Resources フォルダに置かれた" 伝統的な" リソースファイルはリソーステーマのコマンドを使用して、明示的にコンポーネントにロードしなければなりません。

1 つ以上のコンポーネントを含むホストデータベースでは、ホストデータベースと同様それぞれのコンポーネントが固有のリソースチェーンを持っています。リソースは異なるデータベース間で分離されます。コンポーネントA のリソースにコンポーネントB やホストデータベースからアクセスすることはできません (共有及び非共有オブジェクト参照)。

**コンポーネントのオンラインヘルプ**

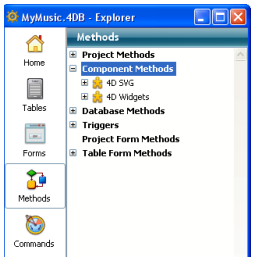
コンポーネントにオンラインヘルプを追加できるようにするために、特別なメカニズムが実装されました。原理は4D データベースに提供されているものと同じです (付録 A: カスタムヘルプファイルを割り当てる参照) :

- コンポーネントヘルプは拡張子が.htm, .html または (Windows のみ) .chm で提供される。
- ヘルプファイルはコンポーネントのストラクチャファイルと同階層に置かれ、ストラクチャーと同じ名前である。
- 結果、このファイルは自動でアプリケーションのヘルプメニューに、" ヘルプ:ヘルプファイル名 " のタイトルでロードされます。

# コンポーネントとホストデータベースの相互作用

### コンポーネントの表示

ホストデータベースにコンポーネントがインストールされると、ホストデータベースのエキスプローラーのメソッドページ内、コンポーネントメソッドテーマにコンポーネント名が表示されます。共有プロジェクトメソッドが階層リストとして表示され、コンポーネントがインタプリターであれば、プリビューエリアに内容が表示されます。




共有メソッドの宣言についてはプロジェクト メソッドの共有を参照してください。

### 共有及び非共有オブジェクト

コンポーネントで定義される特定のタイプのオブジェクトは、自身の実行空間で展開されます。これによりホストデータベースや他のコンポーネントのオブジェクトとのコンフリクトの可能性を取り除いています。これらのオブジェクトは" 非共有または" 分割された" オブジェクトと呼びます。例えば変数は分割されたオブジェクトであり、あるコンポーネントにおける倍長整数タイプの<>Myvar 変数と、ホストデータベースや他のコンポーネントで使用されるテキストタイプの<>Myvar 変数とを同時に使用することができます。

他方のオブジェクトはホストデータベースとコンポーネントで同じ実行空間を共有します。これらのオブジェクトを使用する際にはより注意が必要となりますが、他方ホストデータベースとコンポーネント間で通信を行うことが可能です。これらのオブジェクトは" 共有" または" 分割されない" オブジェクトと呼ばれます。

例えばセットは分割されないオブジェクトであり、コンポーネントが作成した"mySet" を、ホストデータベースで実行される以下のコードで消去することが可能です:

```
CLEAR SET("mySet")
```

#### 非共有オブジェクト

以下は非共有(分割された) オブジェクトです:

- スタイルシート
- ヘルプTips
- 選択リスト
- ライブラリピクチャー
- メニューエディターで作成されたメニューやメニューバー
- " コンポーネントとホストデータベースで共有する" 属性を持たないプロジェクトメソッド
- セマフォー
- プロセス
- 変数 (ローカル, プロセス, インタープロセス)
- システム変数 (OK, Document, など)
- テーブルフォーム
- サブフォームとして公開属性を持たないプロジェクトフォーム
- リソースと開いたリソースファイルの参照

#### 共有オブジェクト

以下は共有( 分割されない) オブジェクトです:

- セット
- 命名セレクション
- 参照を使用した階層リストe (New list、Load list、Copy listまたはBLOB to listコマンドを使用して作成)
- Create menuコマンドから返されるID を使用したメニューやメニューバー
- " コンポーネントとホストデータベースで共有する" 属性を持つプロジェクトメソッド
- ホストデータベースにサブフォームとして公開属性を持つプロジェクトフォーム
- XMLストラクチャー参照
- ファイル参照 (リソースファイルを除く)
- ポインター

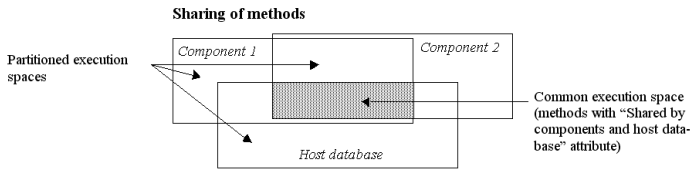
**注:** もちろん、マトリクスデータベース内で見つかった利用できないオブジェクトは、ホストデータベース側で無視されます (利用可能又は利用不可能なオブジェクト参照)。

### プロジェクト メソッドの共有

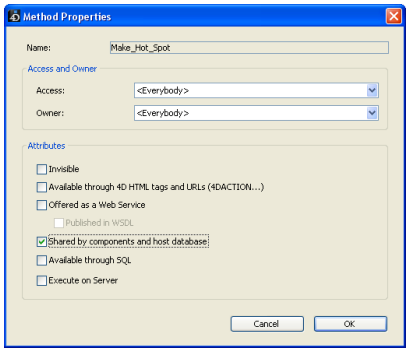
マトリクスデータベースのすべてのプロジェクトメソッドは 、コンポーネントに含まれます。つまり、マトリクスデータベースのプロジェクトメソッドは、すべて自身のコンポーネントから呼び出して実行することができます。

他方、デフォルトで、これらのプロジェクトメソッドはホストデータベースに表示されず、呼び出すこともできません。ホストデータベースでもプロジェクトメソッドを共有するためには、マトリクスデータベース側でそのメソッドを共有されるよう設定しなければなりません。設定することで、それらのプロジェクトメソッドはホストデータベースのエクスプローラのCurrent form tableに表示され、呼び出すことができるようになります ( しかしホストデータベースのメソッドエディタで編集することはできません)。これらのメソッドはコンポーネントのエントリーポイントとなります。

セキュリティのため、デフォルトでコンポーネントはホストデータベースのプロジェクトメソッドを実行することはできません。特定のケースで、コンポーネントがホストデータベースのプロジェクトメソッドにアクセスできるようにする必要があるかもしれません。そうするためには、ホストデータベースのプロジェクトメソッドで、コンポーネントからのアクセスを可能にするよう明示的に指定しなければなりません。




これはメソッドプロパティダイアログボックスの、**コンポーネントとホストデータベースで共有する**で設定します:




メソッド属性の一括設定ダイアログを使用して、複数のメソッドに対し一気にこの属性を設定することもできます (属性一括設定参照)

このオプションの効果は、データベースがどのように使用されるかにより異なります: データベースがコンポーネントとして利用される場合、メソッドはホストデータベースからアクセス可能で、エクスプローラで見ることができます。データベースがホストデータベースの場合、メソッドはコンポーネントから利用可能です。

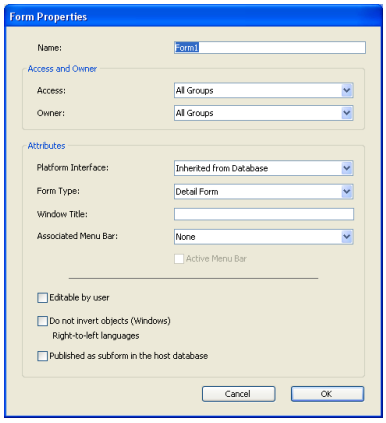
### フォームの共有

マトリクスデータベースのフォームを共有し、ホストデータベース中でサブフォームとして利用できます。

#### サブフォームを公開 (コンポーネント)

コンポーネント側 (マトリクスデータベース側) で、プロジェクトフォームのみを公開設定できます。

ホストデータベース中でサブフォームとしてコンポーネントフォームを選択するためには、フォームプロパティダイアログボックス内で明示的に公開フォームとして定義しなければなりません。:



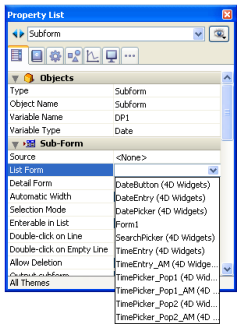

**注:** このダイアログボックスはコンテキストメニューまたはエクスプローラーのフォームアクションメニュー内、**フォームプロパティ...**からアクセスできます (フォームプロパティ (エクスプローラー)参照)。

サブフォームとホストデータベースの親フォーム間の相互作用は開発者が管理します。このメカニズムについてはページサブフォームで説明しています。

#### コンポーネントサブフォームを使用する (ホストデータベース)

ホストデータベース側ではコンポーネント由来のサブフォームをページモードで利用しなければなりません。フォームエディターで親フォームを開きサブフォームオブジェクトを選択したら、プロパティリストのサブフォームテーマから**出力サブフォーム**の選択を解除します。

そしてテーブルメニューから<なし>を選択します。するとコンポーネントから公開されたフォームが詳細フォームメニューにリストされます。フォーム名の後ろには括弧内にコンポーネント名が現れます。このリストからサブフォームとして使用するコンポーネントフォームを選択します。




すると選択されたフォームの新しいインスタンスが即座にフォーム中に作成されます。

### 変数を渡す

ローカル、プロセス、インタープロセス変数は、コンポーネントとホストデータベース間で共有されません。ホストデータベースからコンポーネントの変数、またはその逆の変数にアクセスする唯一の方法はポインターを使用することです。

**配列を使用した例**:

- **ホストデータベース側**:
```
ARRAY INTEGER(MyArray;10)
AMethod(->MyArray)
```
- **コンポーネント側で AMethod プロジェクトメソッドは以下の通りです**:
```
APPEND TO ARRAY($1->;2)
```

**変数を使用した例**:

```
C_TEXT(myvariable)
component_method1(->myvariable)
C_POINTER($p)
$p:=component_method2(...)
```

ホストデータベースとコンポーネント間でポインターを使用して通信を行うには、以下の点を考慮する必要があります:

- Get pointer をコンポーネントで使用した場合、このコマンドはホストデータベースの変数へのポインターを返しません。また逆にこのコマンドをホストデータベースで使用した場合も同様です。

- 新しいコンポーネントアーキテクチャーでは、インタープリターデータベースで、インタープリター及びコンパイル済み両方のコンポーネントを利用できます。(コンパイル済みデータベースでは、コンパイルされたコンポーネントしか使用できません)。この場合、ポインターの利用は以下の原則を守らなければなりません: インタープリター版では、コンパイルモードにビルトインされたポインターを解釈できます。逆にコンパイルモードではインタープリターモードにビルトインされたポインターを解釈することはできません。
  以下の例でこの原則を説明します: 同じホストデータベースにインストールされた2 つのコンポーネントC ( コンパイル済) と I ( インタープリタ)があります:
  - コンポーネントC が定義する変数myCvar があるとき、コンポーネントI はポインター->myCvar を使用して変数の値にアクセスすることができます。
  - コンポーネントI が定義する変数myIvar があるとき、コンポーネントC はポインター->myIvar を使用しても変数の値にアクセスすることはできません。このシンタックスは実行時エラーを起こします。

- RESOLVE POINTERを使用したポインターの比較はお勧めできません。 変数の分割の原則により、ホストデータベースとコンポーネント ( あるいは他のコンポーネント) で同じ名前を持つ変数が存在することができますが、根本的にそれらは異なる内容を持ちます。両コンテキストで、変数のタイプが違うことさえありえます。
  ポインターmyptr1 と myptr2 がそれぞれ変数を指すとき、以下の比較は正しくない結果となるかもしれません:
```
RESOLVE POINTER(myptr1;vVarName1;vtablenum1;vfieldnum1)
RESOLVE POINTER(myptr2;vVarName2;vtablenum2;vfieldnum2)
If(vVarName1=vVarName2)
  //変数が異なっているにもかかわらず、このテストはTrue を返します
```
  このような比較を行うためには、ポインターを比較しなければなりません:
```
If(myptr1=myptr2) //このテストはFalse を返します
```

### ホストデータベースのテーブルへのアクセス

コンポーネントでテーブルを使用することはできませんが、以下のコマンドをコンポーネントの中で呼び出すことはできます:

DEFAULT TABLE
NO DEFAULT TABLE
Current default table

実際、コンポーネントがホストデータベースのテーブルを使用する必要がある場合に、これらのコマンドはとても便利です。この方法で、ホストデータベースとコンポーネントはポインターを使用して通信を行うことができます。例えば、以下はコンポーネントで実行可能なメソッドです:

```
C_LONGINT($1) //ホストデータベースのテーブル番号
$tablepointer:=Table($1)
DEFAULT TABLE($tablepointer->)
CREATE RECORD //ホストデータベースのデフォルトテーブルを使用
$fieldpointer:=Field($1;1)
$fieldpointer->:="value"
```

**ランゲージコマンドのスコープ**

使用できないコマンドを除き、コンポーネントではすべての4D ランゲージコマンドが使用できます。

コマンドがコンポーネントから呼ばれると、コマンドはコンポーネントのコンテキストで実行されます。ただし EXECUTE METHODコマンドは除きます。このコマンドはコマンドで指定されたメソッドのコンテキストを使用します。またユーザ＆グループテーマの読み出しコマンドをコンポーネントで使用することができますが、ホストデータベースのユーザ＆グループ情報を読み出すことに注意してください ( コンポーネントに固有のユーザ＆グループはありません)。

SET DATABASE PARAMETERとGet database parameterコマンドは例外となります: これらのコマンドのスコープはグローバルです。これらのコマンドがコンポーネントから呼び出されると、結果はホストデータベースに適用されます。

さらにStructure fileとGet 4D folderコマンドは、コンポーネントで使用できるように変更が行われています。

COMPONENT LISTコマンドを使用して、ホストデータベースにロードされたコンポーネントのリストを取得できます。

**デバッグ**

コンパイルされていないコンポーネントを使用すると、コードの内容がホストデータベースの標準デバッガーに表示されます。

デバッガーは分割されたオブジェクトの実行空間を考慮します。カスタムウォッチペインでホストデータベースの変数var1 の値を表示し、同じくvar1 変数を含むコンポーネントのコードを実行しても、表示される値は更新されません。カレントコンテキストの変数値を表示させるためには、変数のインスタンスを別にカスタムウォッチペインに登録しなければなりません。

# 付録

付録 A: カスタムヘルプファイルを割り当てる
付録 B: Mac OSでのコマンドラインインターフェース
付録 C: XLIFFアーキテクチャー
付録 D: 特別な辞書を使用する

# 付録 A: カスタムヘルプファイルを割り当てる

4Dでは各データベースにカスタムヘルプファイルを関連付けることができます。

4Dが提供するヘルプシステムは各4D動作環境 (スタンドアロンデータベースまたはクライアントサーバー、インタープリターまたはコンパイル、4D Volume Desktop統合) で互換性があります。

4Dは2つのヘルプファイルフォーマット、CHMとHTMLをサポートします。これらのフォーマットは業界標準に対応します。

さらにこのヘルプファイルの特定の節をデータベースのフォームに割り当てることができます。

## データベースのヘルプを設定する

### ファイルフォーマット

以下のいずれかのフォーマットを利用できます:

- **コンパイルされたHTMLフォーマット** (".CHM"拡張子): これはWindows標準のヘルプフォーマットです。CHMファイルはOSに統合された*Microsoft® HTML Help Executable* (HH.EXE) アプリケーションで開かれます。これらのファイルはWindows XPとWindows Vista互換です。
  Microsoft® は無料で *HTML Help Workshop* アプリケーションを提供しています。このアプリケーションを使用してコンパイルされた .CHM ファイルを生成できます。
- **HTMLフォーマット**(".HTM"拡張子): このフォーマットを使用すればすべてのプラットフォームで利用できるヘルプファイルを作成できます。ヘルプはWebブラウザーに表示されます。

### データベースにヘルプファイルを割り当てる

ヘルプファイルを生成したら、それをデータベースに関連付け、ユーザーの要求に応じて開かれるようにします (データベースからヘルプを呼び出す参照)。ヘルプファイルをデータベースに関連付けるには:

- ヘルプファイル名はデータベースのストラクチャー名と同じにします。拡張子はフォーマットに応じて".CHM"または".HTM"とします。
- ヘルプファイルはストラクチャーファイルと同階層か、データベースストラクチャーファイルと同階層のResourcesフォルダー内に配置します。

**4D Server:** クライアントマシンからヘルプファイルにアクセスできるようにするには、ヘルプファイルを**Resources**フォルダーに配置します。そうすれば自動でファイルがクライアントに転送されます。

プラグインもヘルプファイルを持つことができ、この場合ヘルプファイルをシングルユーザーでもクライアント/サーバーでも**Plug-Ins**フォルダーに配置しなければなりません。プラグインのヘルプファイルはプラグインと同じ名前で、拡張子が.CHMまたは.HTMでなければなりません。フォルダー形式で提供されるプラグインのヘルプファイルは以下の場所に配置します:
- プラグインのプログラムファイルと同じ場所、つまり /Contents/MacOS や /Contents/Windows
- またはプラグインの /Contents/Resources フォルダー

コンポーネントにカスタムヘルプファイルを含めることもできます。詳細はコンポーネントのオンラインヘルプを参照してください。

### コンテキストヘルプファイルを作成する

コンテキストヘルプの作成は、各フォームにセクション番号を指定することで行います。フォームからヘルプを呼び出した場合、対応するページが表示されます。ユーザーがヘルプを呼び出した場合、フォームと同じIDのヘルプトピックが表示されます。

コンテキストヘルプは以下の形式で作成できます:

- WindowsではCHMフォーマット
- WindowsとMac OS両方で使用する場合HTML

フォームへのID番号の割り当てはフォームプロパティで行います。詳細はヘルプを参照してください。

フォームにID番号を割り当てたら、ヘルプファイルに同じ番号を割り当てます。この作業は使用するフォーマットにより異なります:

- CHMファイルの場合、*HTML Help Workshop* アプリケーションのドキュメントを参照してください。
- HTMLファイルの場合、<a>タグを使用して節を宣言し、番号を割り当てます。

  ```
  <a name="section">
  ```

  例えば:

  ```
   <a name="21">
  ```

  その節へのURLは以下の形式になります:

  ```
  <a href="#section">...</a>
  ```

  例えば:

  ```
  <a href="#21">...</a>
  ```

フォームに指定された節番号が0の場合、またはファイルに対応する箇所が存在しない場合、4Dは最初のページを表示します。

## データベースからヘルプを呼び出す

4Dからはデータベースヘルプファイルを以下2つの方法で呼び出せます:

- **ヘルプ**メニューから**データベース名 ヘルプ**を選択する。この場合ヘルプの最初のページが表示されます。
- アプリケーションモードでフォームが表示されているときに**F1** キーを押す。この場合、ヘルプトピック番号が指定されていれば、対応するページが表示されます。そうでない場合最初のページが表示されます。

**注:** Mac OSでは**F1**キーの動作をカスタマイズ可能です。アプリケーションモードでユーザーが**F1**キーを押したとき:

- "データベース名.HTM"ファイルがデータベースストラクチャーと同階層にある場合、4Dがヘルプを管理する標準のメカニズムが使用されます。ヘルプファイルがデフォルトブラウザーウィンドウで開かれます。
- そうでない場合、4Dは何も行わず、開発者がイベントを処理できます。つまり完全にカスタマイズされたヘルプシステムを実装することができます。

# 付録 B: Mac OSでのコマンドラインインターフェース

Mac OS Xのターミナルを使用してコマンドから4Dアプリケーション (4Dおよび4D Server) を操作できます。この機能を使用するとリモートからデータベースを開いたり閉じたりでき、特にWebサーバーとして動作する4Dの管理に便利です。

## 基本情報

Mac OS Xターミナルを使用して4Dコマンドを実行できるようにするために、パッケージ内の実行可能アプリケーションバイナリが格納されたフォルダーにアクセスしなければなりません (Contents/MacOS パス)。
例えば4Dパッケージが *MyFolder* フォルダーにある場合、パスは以下のようになります:

```
/MyFolder/4D.app/Contents/MacOS/4D
```

**注:** アプリケーションアイコンをターミナルにドラッグ＆ドロップしてパスを取得することもできます。

コマンドラインの最後に"&"文字を渡して、アプリケーションが別プロセスのバックグランドタスクとして実行されるよう指定することを推奨します例えば:

```
/MyFolder/4D.app/Contents/MacOS/4D &
```

## コマンドと引数

4Dアプリケーションがサポートするコマンドと引数は以下の通りです:

- **アプリケーションを起動**
  *シンタックス*: Path/ApplicationName
  *例*:
  ```
  4D.app/Contents/MacOS/4D &
  ```
  このコマンドの結果は4Dアプリケーションのダブルクリックと同じです。アプリケーションが実行され、データベース選択ダイアログが表示されます。

- **ストラクチャーファイルを指定してアプリケーションを起動**
  *シンタックス*: Path/ApplicationName StructurePath
  *例*:
  ```
  4D.app/Contents/MacOS/4D /Users/Roger/Databases/MyDatabase/MyDatabase.4DB
  ```
  このコマンドはアプリケーションを起動し、指定されたストラクチャーファイルとカレントデータファイルを開きます。データベース選択ダイアログボックスは表示されません。

- **ストラクチャーファイルとデータファイルを指定してアプリケーションを起動**
  *シンタックス*: Path/ApplicationName -d DataPath StructurePath
  *例*:
  ```
  4D.app/Contents/MacOS/4D -d /Users/Roger /Databases/MyDatabase/MyDatabase.4DD
  /Users/Roger/Databases/MyDatabase/MyDatabase.4DB
  ```
  このコマンドはアプリケーションを起動し、指定されたストラクチャーファイルと選択されたデータファイルを開きます。データベース選択ダイアログボックスは表示されません。

- **アプリケーションを終了**
  *シンタックス*: kill -s INT process_ID
  *例*:
  ```
  kill -s INT 323
  ```
  このコマンドの動作は**終了**コマンドの選択と同じです。プロセスはひとつづつ終了され、キャッシュがフラッシュされ、アプリケーションが終了します。
  **注:** 4Dアプリケーションのプロセス番号は ps や top などのコマンドで取得できます。

- **ヘルプ**
  *シンタックス*: Path/ApplicationName -h
  *例*:
  ```
  4D.app/Contents/MacOS/4D -h
  ```
  このコマンドはMac OS Xターミナルで使用可能なコマンドの説明を表示します。

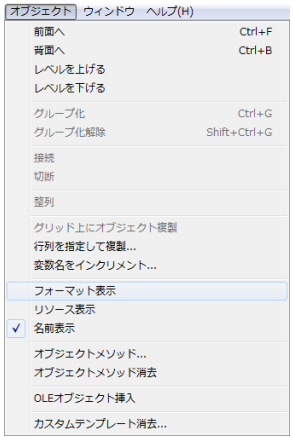
- **カレントセッションでないユーザーセッションでアプリケーションを起動**
  *シンタックス*: Path/ApplicationName -u UserName -g GroupName
  *例*:
  ```
  4D.app/Contents/MacOS/4D -u john -g accnt
  ```
  このコマンドは指定したユーザーセッションで強制的にアプリケーションを起動します (デフォルトではカレントユーザーセッションが使用されます)。

# 付録 C: XLIFFアーキテクチャー

4Dはテキストやタイトルのローカライズのために、XLIFF標準をサポートします。この技術は4Dアプリケーション内部でも使用され、4D開発者やプラグイン開発者もカスタマイズされたアプリケーションやプラグインで利用できます。

使用する言語に合わせて動的にインターフェース言語が変えられる4Dアプリケーションを設定する原則は以下の通りです。言語により変更されるすべての要素 (テキスト、ラベル、ピクチャー) をファイルとしてアプリケーションの外側、**Resources**フォルダーに格納します。フォームやメニューの中で、ラベルやピクチャーは参照として指定されます。アプリケーションが実行されると、言語コンテキストに応じてこれらの要素は外部から表示すべきリソースを取得します。XLIFF標準はラベルの記述と参照方法を規定します。

フォームエディター内ではスタティックエリアを編集モードにするか、**オブジェクト**メニューから**フォーマット表示**または**名前表示**を選択すると、実際の内容を見ることができます。:




**互換性に関する注記:** XLIFFの他に、4Dは現時点で以前のシステムであるSTR#リソース参照もサポートしています。しかしこのシステムは廃止予定です。両方の方法を同時に使用することができます。今のうちにXLIFFを使用した方法に変更してください。

### XLIFFとは

XLIFF (*XML Localization Interchange File Format*) はローカライズ処理のための標準です。XMLを使用して翻訳元テキストと翻訳済みテキストの対応を記述します。

実際XLIFF標準はリソースベースのローカライズシステムに代わる新しい標準です。このファイルを管理するためのツールがいくつか存在します。

XLIFF標準に関する詳細は公式XLIFF 1.1仕様書を参照してください:
http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm

**警告:** XLIFF標準は大文字/小文字を区別します。

### 4DデータベースからXLIFF文字列を呼び出す

#### 参照の場所

4Dでは以下の場所でXLIFFを使用できます:

- フォームエディターのプロパティリストの"オブジェクト名"フィールド
- エクスプローラーからアクセスできるフォームプロパティダイアログボックスの"ウィンドウタイトル"フィールド
- ボタンやチェックボックスタイトル、スタティックテキストなど
- メニューエディター (*ObjectName* シンタックスを除く、後述)
- ヘルプTipsエディター (*ObjectName* シンタックスを除く、後述)

**注:**

- XLIFF参照が認識されるためには、テキストの先頭に記述されなければなりません。
- ヘルプTipエディターにXLIFF参照を表示することはできません。XLIFFに基づくヘルプTipを使用するには、プロパティリストのヘルプメッセージに直接XLIFF参照を入力します。

### 参照のシンタックス

4Dでは以下のシンタックスを使用してXLIFFを参照できます:

- **:15115,3** シンタックス
  XLIFFファイルでこのシンタックスを使用する場合、最初の値 (この例では15115) は**group**要素の**id**属性値に対応します。.
  二つ目の値 (この例では3) は**trans-unit**要素の**id**属性値に対応します。
  **互換性に関する注記:** このシンタックスは前世代のSTR#リソースに基づくものと同じです。このとき":xxxx,yyy"においてxxxxはSTR#リソース番号でありyyyは要素番号でした。例えば":15115,3"はSTR#リソース15115の3番目の文字列を使用することを意味しました。このシンタックスがサポートされるため、変換されたデータベースにおいてリソースベースのシステムが動作し、徐々にXLIFFベースのシステムに置き換えられます。4Dはまずロードされたxliffファイルの中から15115,3に対応するリソースを探します。もし見つからなければ開かれたリソースファイルの中で対応するリソースを探します。このメカニズムを使用すれば既存のSTR#リソースに基づくシステムにXLIFFベースのシステムを統合することが可能です。両方のシステムを同時に利用できます。いずれの場合もロードされたXLIFFファイルがリソースファイルよりも優先されます。

- **:xliff:OKButton** シンタックス
  この新しいシンタックスはロードされたXLIFFファイルでのみ利用できます。この場合参照された名前 (この例では*OKButton*) は**trans-unit**要素の**resname**属性に対応します。

- **ObjectName** シンタックス (プロパティリストの"オブジェクト名"フィールド)
  このシンタックスの目的はフォームの翻訳を容易にすることにあります。他のシンタックスと異なり、これはフォームでのみ使用できます。動作の原理はXLIFFファイルの**trans-unit**要素を、**resname** 属性にそれぞれテーブル名とフォーム名を含む2つの**group**要素で挟むことにあります。
  - 例 (テーブルフォーム)
    [Clients]テーブルの"Form1"フォームにある、オブジェクト名"SaveButton"のボタンタイトルを翻訳するには、XLIFFファイルに以下のように記述します:

    ```
    <group resname="[Clients]">
    <group resname="Form1">
    <trans-unit resname="SaveButton">
    ...
    </trans-unit>
    </group>
    </group>
    ```

  - 例 (プロジェクトフォーム)
    プロジェクトフォームの場合、テーブル名は[ProjectForm]と記述します:

    ```
    <group resname="[ProjectForm]">
    <group resname="Form1">
    <trans-unit resname="SaveButton">
     ...
    </trans-unit>
    </group>
    </group>
    ```

データベース中で異なるシンタックスを使用する場合、XLIFFファイル中で対応するリソースを検索する優先度は以下の通りです:
1. ":15115,3" シンタックス
2. ":xliff:OKButton" シンタックス
3. "MyTitle" シンタックス

"Resources"テーマのGet indexed stringとSTRING LIST TO ARRAYコマンドはXLIFFドキュメントを利用できます。

XLIFF参照またはカレントの言語に従い翻訳されたタイトルをフォームエディターで表示するには、オブジェクトメニューから**名前表示**または**リソース表示**を選択します。詳細はスタティックテキスト中で参照を使用するを参照してください。

### カスタマイズしたXLIFFファイルの配置

アプリケーションのローカライズのためにXLIFFアーキテクチャーを使用する場合、有効なXLIFFファイルを必要なだけ作成し、データベースの**Resources**フォルダーにコピーします。

データベースのResourcesフォルダーに関する詳細はリソースフォルダの管理を参照してください。

使用するパスは以下の通りです:

- Windows:

***MyDatabase*\Resources*{Lang}*.lproj*{MyJALoc}*.xlf**

- Mac OS:
  ***MyPackage*:Resources:*{Lang}*.lproj:*{MyJALoc}*.xlf**

上記において:

- *MyDatabase* はデータベースファイルを格納したフォルダーで、*MyPackage* はMac OSのデータベースパッケージです。
- *{Lang}*.lproj には Lang 言語用のXLIFFファイルを配置します。Lang 部分は後述の国際標準に従った名称を使用しなければなりません。例えば日本語の場合、フォルダー名は *ja.lproj*です。

4Dはデータベースのカレント言語に対応したXLIFFファイルを自動でロードします。データベースのカレント言語を決定する際、4Dは以下の順番でデータベースの**Resources**フォルダー内を検索します:

1. システム言語 (Mac OSでは環境設定で言語の検索順を設定できます。4Dはこの設定を使用します。)
2. 4Dアプリケーションの言語
3. 英語
4. ここまでで検索できなかった場合、**Resources**フォルダーで最初に見つかった言語がロードされます。

Get database localizationコマンドを使用するとデータベースに指定されたカレント言語を知ることができます。

言語のバリエーションが使用されていて、XLIFFファイルでそれが利用できない場合、それにもっとも近い言語が使用されます。

XLIFFファイルの名前 (例では"MyJALoc"部分) は自由に決定できます。ただし拡張子を".xlf."にします。同じ言語フォルダー内に複数のXLIFFファイルを配置できます。それらはファイル名の順番にロードされます。

#### .lprojフォルダー名

".lproj"フォルダーの名前は後述する標準に従わなければなりません。4Dは標準に従ったフォルダーを以下の順番で検索対象とします。:

1. **言語-地域コード**
   この組み合わせでは言語コード (ISO639-1) + ハイフン + 国コード (ISO3166) または 4文字のスクリプトコー (ISO15924) となります。
   たとえばカナダフランス言語圏では"fr-ca" (*fr-ca.lproj*) となります。
2. **ISO639-1**
   この標準では各言語を2文字で表します。例えば日本語は"ja" (*ja.lproj*) です。
   参照: http://www.loc.gov/standards/iso639-2/php/English_list.php
3. **レガシ名**
   この用法では言語名を英語で表記します。例えば日本語の場合"japanese" (*japanese.lproj*) です。
   **注:** 最初の二つの記法はMac OS 10.4以降でサポートされています。それより前のMac OSではレガシ名のみがサポートされています。

4Dがサポートする言語コードの要約は言語コードで提供されます。

**注:** 同じ原則がXLIFFファイル中の"target-language"タグにも適用されます。このタグの値を注意深く設定しなければなりません。ファイルが"fr-ca.lproj"フォルダー内に存在するとしても、"target-language=fr"と記述されていた場合、そのファイルは"fr"として扱われます。

#### カスタマイズしたXLIFFファイルのリロード

アプリケーションをローカライズする際、起動時にロードされたXLIFFファイルは動的に再ロードされます。つまりXLIFFファイルを編集すると、その変更は即座に4D内に反映されます。このリロードは4Dがアプリケーションの最前面になったときに、XLIFFファイルの更新日時が前回のロード時と異なる場合に発生します。カレントフォームも同時に再ロードされます。

### 言語コード

4Dがサポートするの言語コードは以下の通りです。

| 言語 | ISO639-1 | “レガシ” | ISO3166 / ISO15924 |
|---|---|---|---|
| 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 | hant |
| CHINESE_SIMPLIFIED | zh | chinese | hans |
| 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 | |
| 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_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 | |
| 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 |
| 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 | |

# 付録 D: 特別な辞書を使用する

4Dにはスペルチェックで受け入れるカスタマイズされた単語のリストを使用するメカニズムがあります。これにより専門分野、アプリケーションの目的、会社の用途に応じて特別な辞書を利用することができます。

主たる言語ごとに複数の特別辞書を使用できます。これは4Dおよび4D Writeで動作します。

4Dのスペルチェックに関する詳細はスペルチェックを参照してください。

## インストール

特別辞書を指定した言語に追加するには、テキストフォーマットのファイルを**Spellcheck**フォルダー内の主たる言語サブフォルダー内に配置するだけです。ファイル名は何でも構いませんが、拡張子は".txt"でなければなりません。

言語サブフォルダーは4Dの以下の場所に作成されます:

- Windows: (*4D Application folder*)\Resources\Spellcheck\
- Mac OS: (*4D Package*):Contents:Resources:Spellcheck:

サブフォルダーの名称は主たる言語に対応します。変更することはできません:

- English
- French
- German
- Spanish
- Norsk

各フォルダーにはデフォルトで単語ファイルが含まれています。このファイルに単語を追加したり、他のファイルを追加したりできます。テキストエンコーディングはUTF-8でなければなりません。

### クライアント/サーバー

スペルチェックファイルは4Dアプリケーション内に格納されています。特別辞書をクライアント/サーバーモードで利用したい場合、先に説明した箇所にその辞書をそれぞれインストールしなければなりません。

## ファイルの内容

特別辞書の内容は改行で区切られた単語のリストです。例えば:

```
4D
Server
Desktop
```

これに単語 (例えば: *boogie*) を追加したり、ハイフンで連結された複合単語 (例: *boogie-woogie*) を追加できますが、スペースで区切られた単語の並びは追加できません (例: *Mark Smith*)。

テキストファイルの内部フォーマットはWindowsとMac OSで異なります (特に改行コード)。そのためプラットフォームごとに特別辞書を作成する必要があります。テキストエディターには改行コードを変換できるものがあります。

## ロードと利用

特別辞書は標準辞書と同様、アプリケーションの起動時にロードされます。辞書のカレント言語はアプリケーション言語に基づきます。

セッション中、SPELL SET CURRENT DICTIONARYコマンドを使用してカレント言語を変更できます。この場合特別辞書が対応する言語フォルダーに存在すれば、それもロードされます。

利用中、標準辞書と特別辞書の間に単語の処理に関する違いはありません。