



# 4D データベースを 4D v11 SQL に変換する

4D データベースを 4D v11 SQL に変換する

## 必要な動作環境

Mac: iMac 以上 / Mac OS X 10.4.5

512 MB RAM 以上 - 1 GB 推奨

Certified バージョン: Mac OS X 10.4.10

Windows: Pentium II minimum / Windows XP

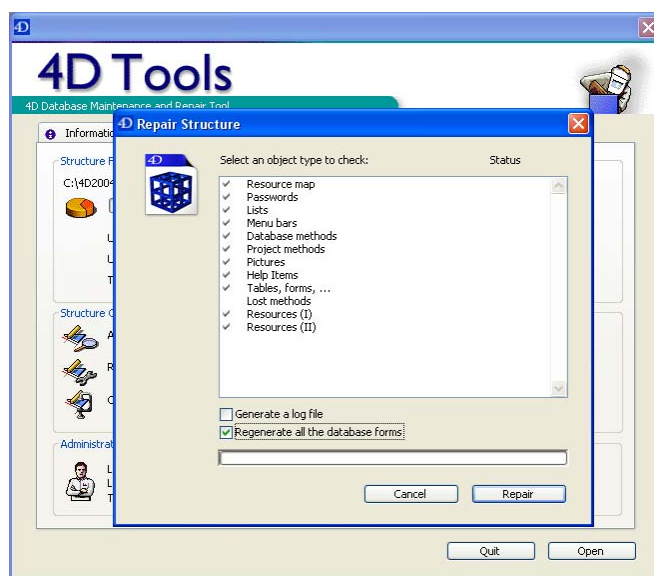
512 MB RAM 以上 - 1 GB 推奨

Certified バージョン: Windows Server 2003 R2, Windows XP SP2, Windows VISTA

## 以前のデータベースを 4D v11 SQL に変換

バージョン 6.5 (さらには 6.0)、2003 または 2004 データベースを直接 4D v11 SQL 用に変換できます。

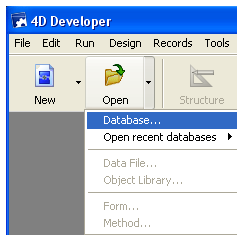
ただし、変換を行う前に、元のバージョンに対応する 4D Tools を使用して、データとストラクチャの検査を行うことをお勧めします。まだ"すべてのデータベースフォームを再生成"オプションを使用していない場合は、このオプションを使用してフォームを再生成してください。このオプションは 4D Tools バージョン 2004 でのみ有効です:



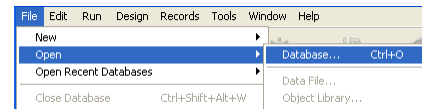
4D v11 SQL を起動します:



4D のメニューバーが表示されます。ツールバーの開くを選択してデータベースを開きます:

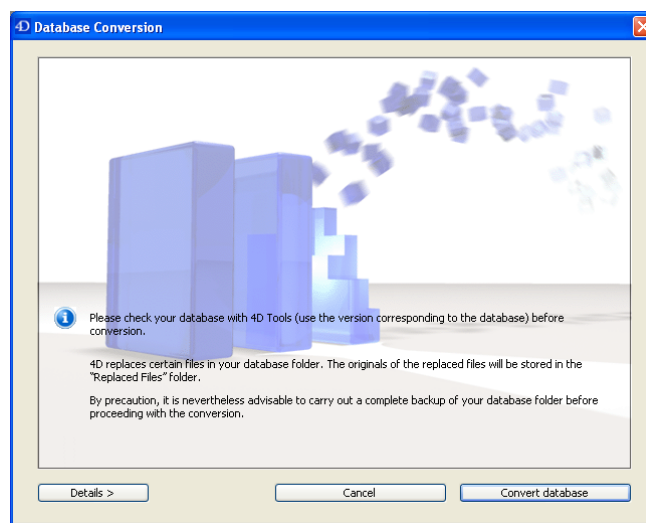


または開くメニューを使用します:



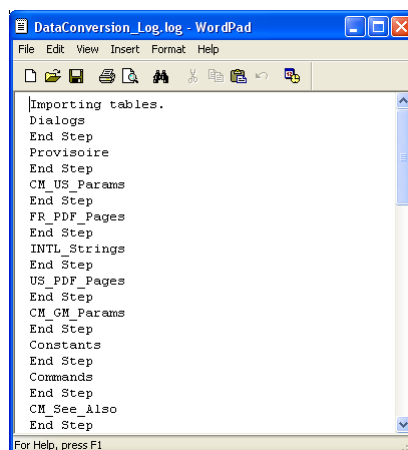
変換する4DB ファイルを選択します。すると変換ウィザードが表示されます。

## 変換ウィザード



変換ウィザードが提供されます。ウィザードは元のデータベース (ストラクチャ、データ、ログファイル、および空のログファイル) のコピーを、"Replaced files (Conversion)" というフォルダに作成します。

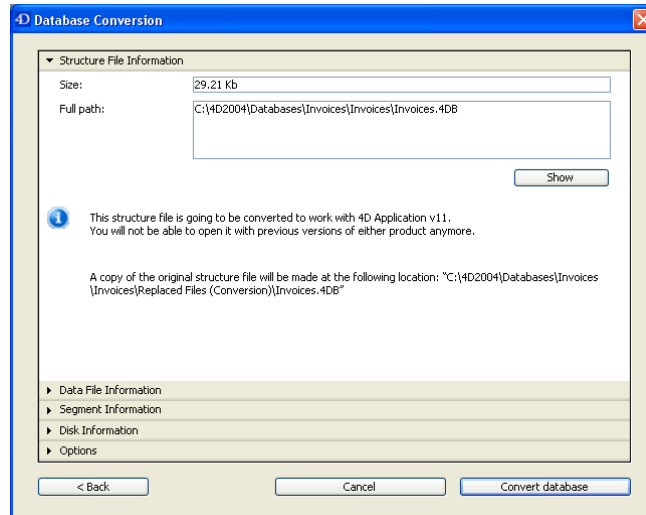
変換されたストラクチャファイルと同階層には、"DataConversion\_Log.log" という名前のファイルが作成され、変換のステップが記録されます。



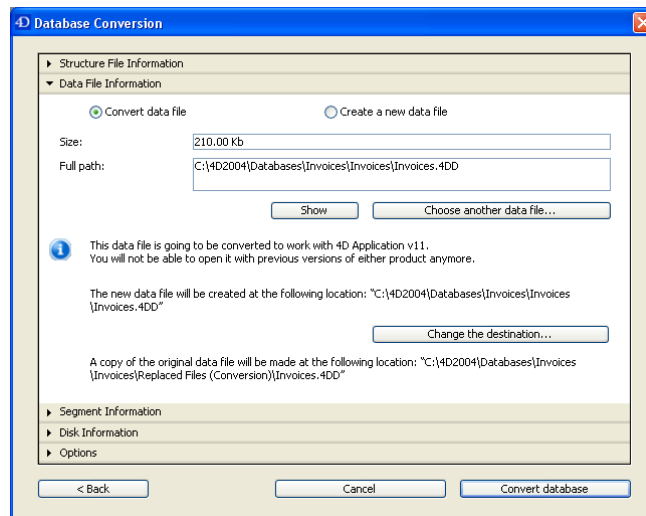
このファイルには、以下のエラーが記録される可能性があります:

1012: 1 レベルを超えるサブテーブルを持つデータベースを変換しました。

## ストラクチャファイル情報:

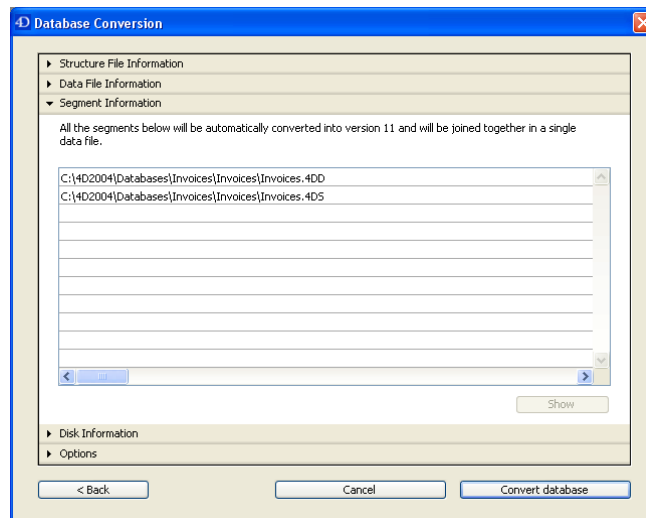


## データファイル情報:



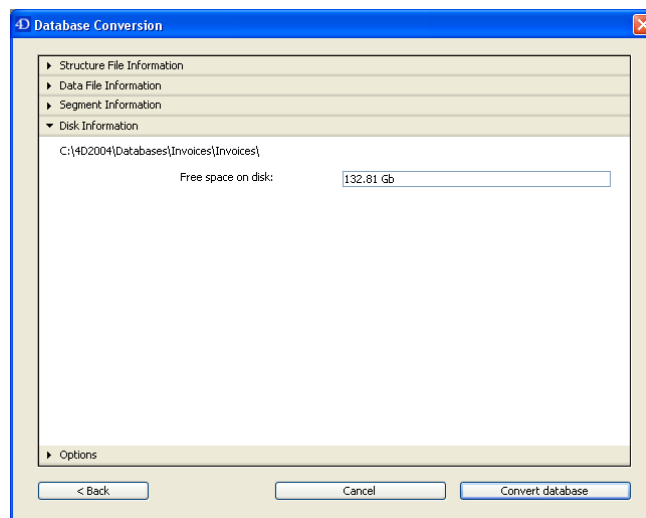
新しいデータファイルを作成できます。

## セグメント情報:



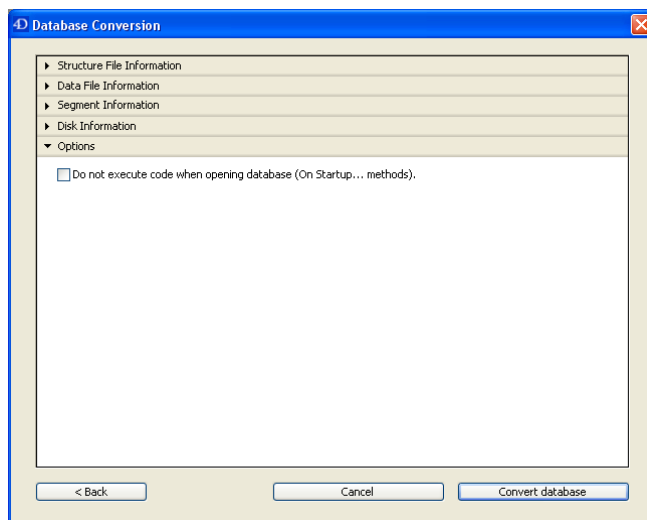
4D v11 SQLにはデータセグメントが存在しません。変換を行うと、すべてのセグメントはひとつにグループ化されます。すべてのセグメントがそろっていないければ、変換を行うことはできません。

## ディスク情報:



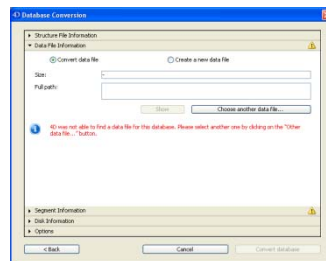
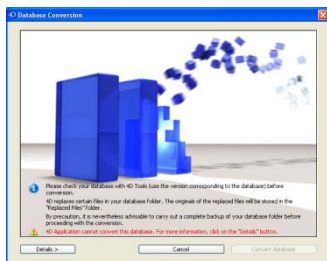
## オプション:

ここでは変換したデータベースを初めて開く際にコードを実行しないよう設定することができます:



このオプションは重要です。このオプションを選択すると、変換したデータベースを最初に開く際に、"On Startup"データベースメソッドを実行しません。

問題が発生すると、エラーメッセージが表示され、問題の箇所を示します。例えば次のスクリーンショットはデータファイルが見つからない場合のエラーを示しています。



## 変換時に問題がある場合

- 新しいデータファイルを生成してください。
- 変換オプションの"データベースを開く際にコードを実行しない"を選択してください。
- Maintenance & Security Center (MSC) を開き、レコードとインデックスを検査して、さらにアプリケーションを検査してください。
- 必要なプラグインをインストールしてください (Plugins と Win4DX/Mac4DX フォルダ)。
- インタプリタモードとコンパイルモード両方をチェックしてください。

注: 4D v11 SQL のインタプリタモードは 4D 2004 に比べ、より厳密に動作します。

## 変換時の動作:

### 新しいファイル

4D v11 SQL に変換後、4D データベースにはいくつかのファイルが追加されます:

- DatabaseName.4DIndy: ストラクチャのインデックス
- DatabaseName.4DIndx (オプション): データのインデックス
- DataConversion\_Log.log: このファイルには変換のステップおよび発生したエラーが記録されます。このファイルに問題が報告されている場合は、MSC を使用してデータベースを修復してください。

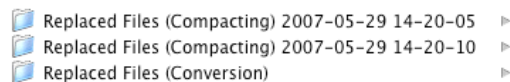
注: 4D v11 SQL で作成されたデータベースは .4dbase フォルダに格納されます。

Mac OS 上では、データベースはパッケージとして表示されます。

### 新しいフォルダ

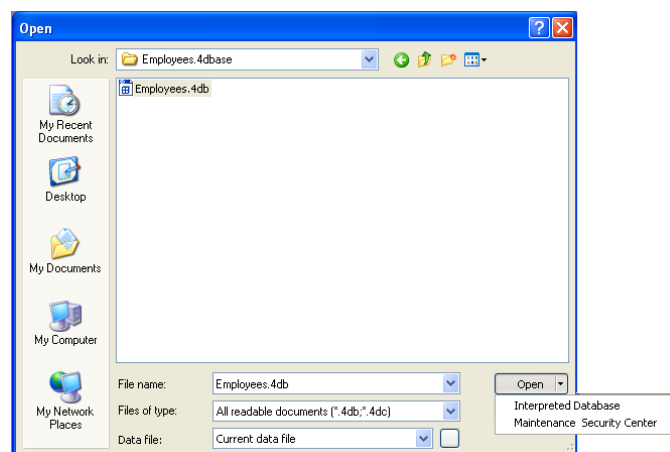
- Replaced files:

いくつかの操作により、Replaced files という名前のフォルダが作成されます: これらのフォルダには、変換前や圧縮前、または修復前のもとのファイルのコピーが納められます。

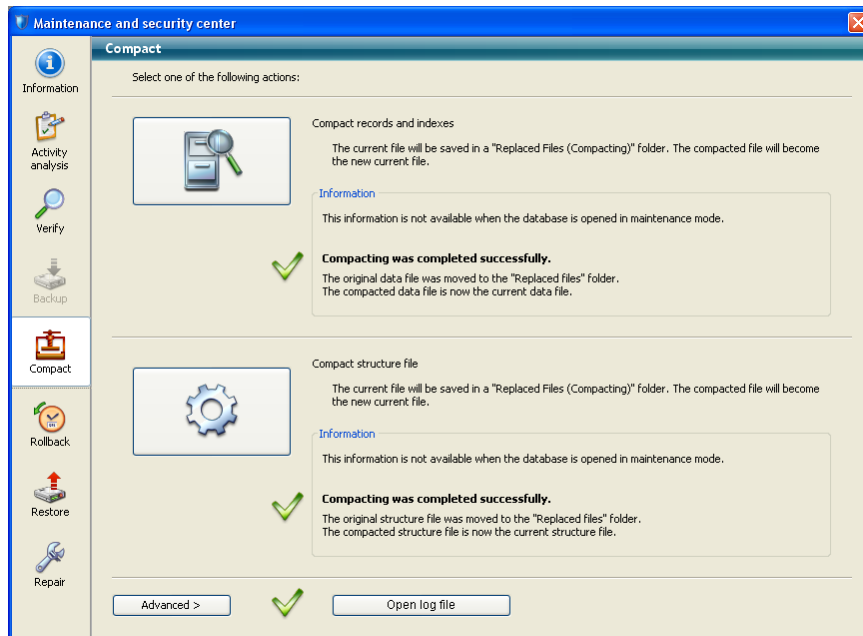


- Macros v2 (マクロを参照)
- Resources (リソースを参照)

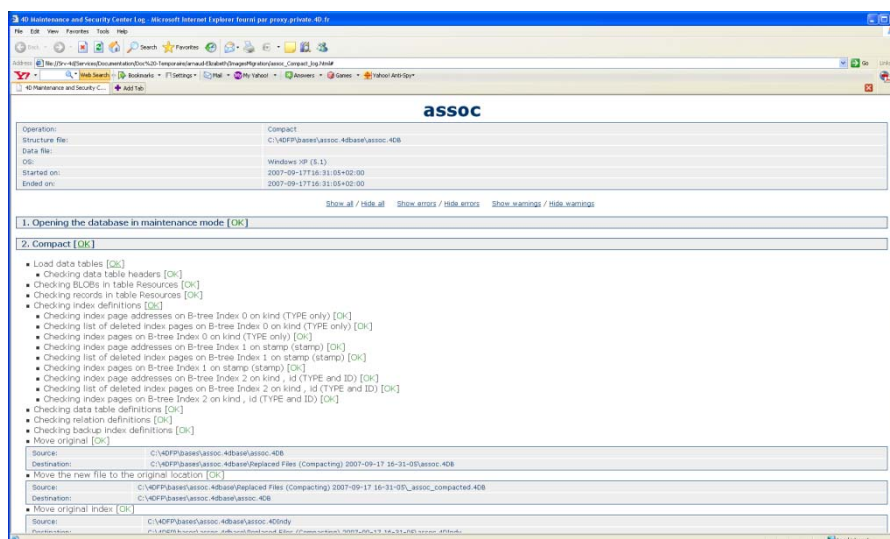
## 変換後に Maintenance & Security Center を使用する



Maintenance & Security Center (MSC) は 4D Tools を置き換えるものです。MSC を使用するには、4D Developer を起動します:



ここではストラクチャやレコードの修復や圧縮を行うことができます。下図はログファイルを開くをクリックしたときに表示されるログファイルの例です:





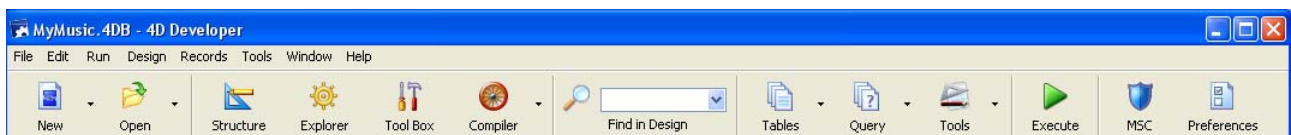
## **変換後に消去されるもの:**

- デバッガの設定  
"起動時にデータベースを公開"のチェックは外されます。変換前にチェックされていても、変換後はチェックされていません。
- 再帰リレーション、特にフィールド自分自身へのリレーションは取り除かれます。
- 1レベルを超えるサブテーブル

## **サポートされないもの**

### **ユーザモード**

デザインモードとユーザモードは統合され、デザインモードでデザイン環境とレコードにアクセスできるようになりました:



## **4D リソースとシステムリソースへの直接アクセス**

Apple 社により推奨される仕様と最近の Mac OS の実装に準拠して、本来の意味でのリソースの概念を使用しないよう、徐々に変更を行います。開発者によっては 4D の内部的なリソース (例えば月の名前やランゲージコマンドなど) を利用している場合があるかもしれませんが、このような方法は今バージョンより明確に禁止されます。ほとんどの場合、4D の内部リソースを使わずとも、定数やランゲージコマンドなど他の方法が利用可能です。

今バージョンからは、コマンドもしくは<:15000,5>シンタックスいずれの方法を用いるかに関わらず、システムや 4D アプリケーションのリソースを呼び出すことは禁止され、またそれはできません。

以下のコードはサポートされません:

```
$Structure :=Structure file
```

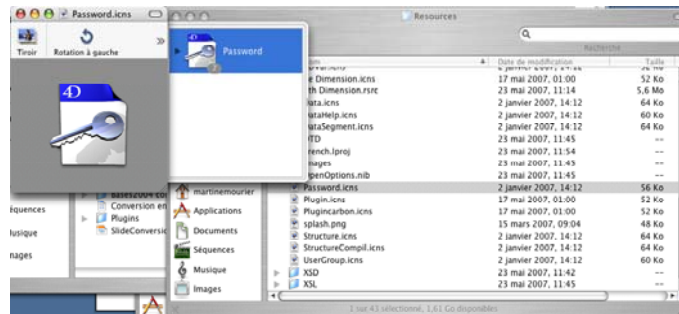
```
$Resources :=Open resource file ($Structure)
```

このコードにおいて、4D v11 SQL では、**Open resource file** コマンドは常に 00:00:00 を返します。

これらのリソースを利用するデータベースにおいて、これらの変更の影響を和らげるために、代替のメカニズムが実装されました。

互換性を保つために、いくつかの変更を行うことで、以前のメカニズムを引き続き動作させることができます。

- リソースファイルが存在する場合、それらは依然として 4D によりサポートされます。また一連の開かれたリソースファイルに関する原則も有効です。一連の開かれたファイルには.rsr と.4dr、および 4D コマンドにより開かれたリソースファイルが含まれます。しかし 4D のリソースにアクセスすることはできません。これは明確に禁止されます。
- この変更の影響を緩和するために、頻繁に使用するリソースについては外部に置きます。



このリソースを利用するには、それらをコピーすることをお勧めします。

## 代替のシステム:

- 必要なファイルはすべて"Resources"という名前のフォルダに置きます。このフォルダはストラクチャと同階層に作成する必要があります。

しかしながら、変換されたデータベースにおいては、すべての 4D 内部リソースの呼び出しを変更または取り除くことを強くお勧めします。

注: 4D v11 SQL で作成されたデータベースには、.RSR と.4DR ファイルは存在しません。

## 4DK# - STR# - cicon - キーボードショートカット - マクロ...

あなたのリソースを取り出すために、"4DPop Migration"コンポーネントが無料で、ダウンロードにて提供されます。このコンポーネントを利用すると以下のリソースを取り出すことができます:

- あなたの STR#。これは.XLIFF ファイルに変換されます。
- .rsr ファイルから取り出されたあなたの 4DK#定数。これは User\_Contants.bundle プラグインに変換されます。
- あなたのキーボードショートカット。この設定は"Extensions"フォルダ内の 4DShortcuts.xml ファイルに書き出されます。
- あなたの v2004 用マクロ。マクロを UTF-8 (クロスプラットフォーム) に変更し、禁止された文字 (// ---- <> 等) を変換します。
- その他

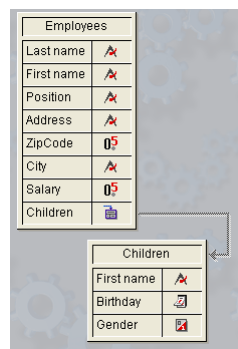
このコンポーネントに関する詳細はお問い合わせください。

## サブテーブル

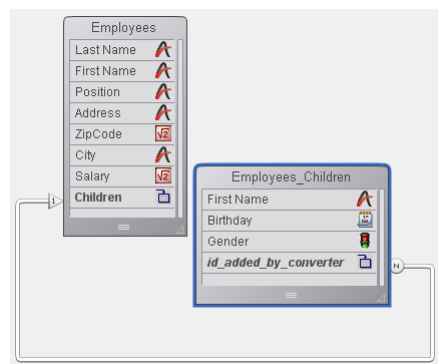
4D v11 SQL では、サブテーブルを作成することができません。

データベースを v11 に変換すると、第一レベルのサブテーブルは、特別なりレーションがつけられた標準のテーブルに変換されます。警告: 第二レベル以降のサブテーブルは削除されます。

例えば以下のようなストラクチャがあります:



これは以下ようになります:



サブテーブルは N テーブルとなり、オリジナルのテーブルは 1 テーブルになります。

新しく作成されたテーブルの名前は ParentTableName\_SubTableName ですが、31 文字で切り取られます。

- N テーブルには主キーとして、id\_added\_by\_converter という名前のフィールドが追加されます。
- N 対 1 リレーションは、SubtableName\_id\_added\_by\_converter、
- 1 対 N リレーションは、id\_added\_by\_converter\_subtable\_1\_Children となります。

サブテーブルが既存のテーブルに追加されるため、v11 のテーブルの順番は v2004 のそれと異なります。

データベースにサブテーブルが含まれる場合、以下の点に留意してください:

- 4D v11 SQL に移行後、テーブル数は増加します。
- 1 レベル以上のサブテーブルがデータベースに含まれる場合、1 レベル以外のサブテーブルは削除されます。この場合、エラー1012 が 'DataConversion\_Log.log' に記録され、また変換後エラー結果ウィンドウにこのエラーが表示されます。
- 変換によって作成された特別なリレーションを削除しない限り、サブレコードに関するコマンドは動作します。しかし一度この特別なリレーションを削除すると、もう元には戻せません。再度リレーションを貼るには標準のリレーションを使い、サブレコードコマンドをテーブルコマンドに置き換える必要があります。

テーブル番号を使用していたり、リレーションを削除した場合は、コードの書き直しが必要となります。

またいくつかのコマンドの動作が変更されたことに留意してください:

- **DUPLICATE RECORD** はサブレコードを複製しません。
- **SEND RECORD** はサブテーブルを送信しません。
- **RECEIVE RECORD** はサブテーブルを受信しません。

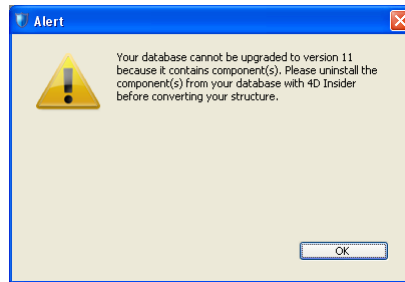
他方、今バージョンからは、**Field** と **Field name** コマンドはポインタ参照が使用された場合、サブフィールドの番号や名前を返します (メインテーブルのそれではなく)。

## **旧来のコンポーネント**

コンポーネントのアーキテクチャは大きく変更されました。コンポーネントは通常のスストラクチャファイルで構成され、コンパイル済みまたはインタプリタで、パッケージの形 (4dbase フォルダ) で提供されます。コンポーネントを使用するには、スストラクチャと同階層の "Components" フォルダにコンポーネントを置きます。

4D v11 SQL のコンポーネントではテーブルを使用することはできません。(4D v11 SQL で作成されたデータベースにはデフォルトの Table1 が含まれず、またテーブルに属さないプロジェクトフォームが利用できます。)

データベースに以前のバージョンのコンポーネントがインストールされている場合は、変換ができない旨の警告が表示されます:

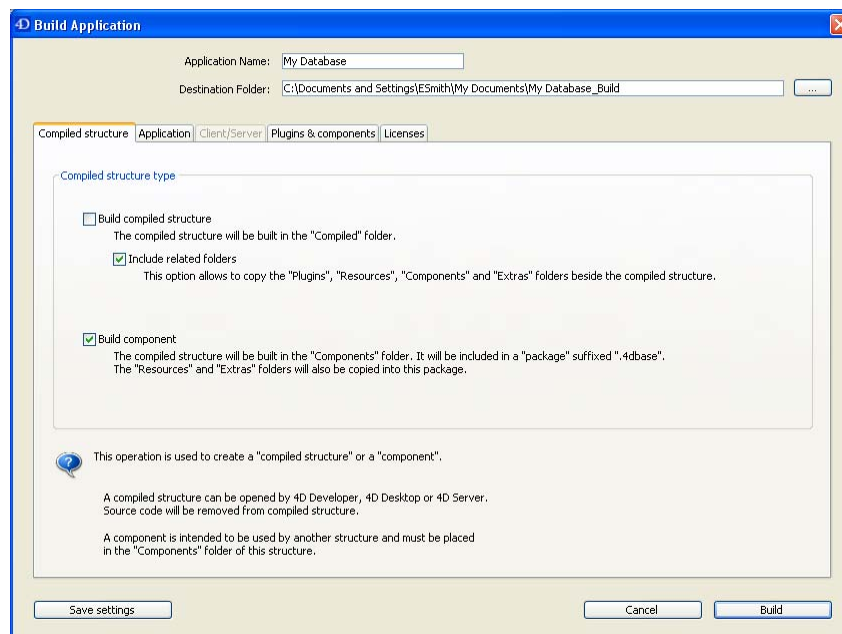


以前のバージョンのコンポーネントは、変換前に、元のバージョンの 4D Insider を使用してアンインストールしなければなりません。

注: 4D v11 SQL コンポーネントでは、以下のオブジェクトを使用することはできません:

テーブルやフィールド、テーブルフォームとそのフォームメソッド、ユーザフォーム、データベースメソッド、およびトリガ。

4D データベースがコンポーネントとなります: .Adb、.A4c または .A4base。プロフェッショナルアプリケーションでは.A4base を推奨します。この形式はアプリケーションビルドダイアログボックスで作成できます:

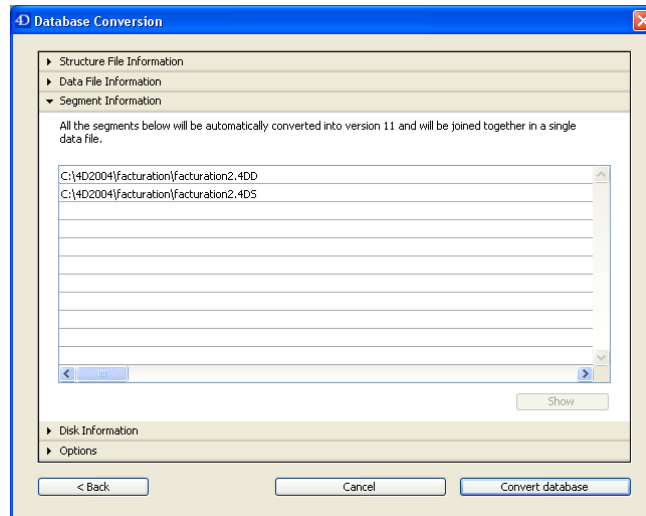


## データセグメント

データファイルのサイズは実質的に制限がなくなります (システムによる制限を除く)。データセグメントを作成したり利用したりすることはできません。以前のセグメントは変換時にひとつにグループ化されます。

変換時にはディスクに十分な空き容量があることを確認してください。

セグメントが失われていると、変換ウィザードはセグメントを選択するよう求めます。見つからない場合、変換を行うことはできません。



## **4D 2003/2004 マクロ**

4D v11 SQL は新しい"Macros v2"フォルダからマクロをロードします。このフォルダはアクティブな 4D フォルダ、ストラクチャと同階層、またはコンポーネントのマクロの場合は"Component"フォルダに置かれません。

4D バージョン 2003 や 2004 で使用するマクロは、4D v11 と互換性がありません。v11 のマクロは XML 標準に従っている必要があります:

```
< ?xml version= « 1.0 »… ?>  
<http://www.4d.com/dtd/v11/Macros.dtd>
```

この情報が書かれていなければなりません。

DTD は以下の場所にあります:

Mac OS: 4D Developer:app:Contents:Resources:DTD:macros.dtd

Windows: 4D Developer¥Resources¥DTD¥macros.dtd

以下の点に留意してください:

// コメント はサポートされません

---- 区切り線 はサポートされません

< や > は &lt; や &gt; にエンコードされなければなりません。

<macros>タグは必須です。

自動の変換メカニズムが実装されています。4D v11 SQL は自動で 4D 2003 と 2004 のマクロを "Macros v2" フォルダにコピーします。宣言部も追加されます。しかしながら XML 標準に適合させるために、変更が必要になる場合があります。

## 新しいコマンド:

GET MACRO PARAMETER

SET MACRO PARAMETER

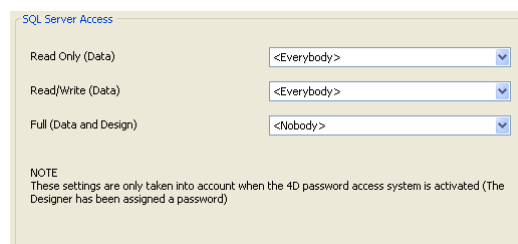
## リソースを使用したログインウィンドウのアイコンのカスタマイズ

この機能は動作しません。このカスタマイズを行うには、.png ファイルを使用します。

使用する PNG ピクチャの名前を "LoginImage.png" とし、ストラクチャと同階層の "Resources" フォルダに置きます。

## テーブルのアクセス権

データの読み込み、追加、保存、および削除に対し、グループを割り当てることはできません。同様にテーブルはオーナーとなるグループを持ちません。これらの制御は変換されたデータベースでは動作しません。今バージョンからは、制御は異なるレベル (フォーム、メニュー、メソッド等)、または (環境設定の SQL サーバ設定など) 特定のオプションで行います。



注: 4D Open による接続を使用したセキュリティホールに留意してください。

## 背景ピクチャのモード選択

モードの選択ウィンドウを使用した、(バックグラウンドフォーマット) ピクチャのバックグラウンド色とフォアグラウンド色の相互作用モードを選択することはできません。この機能は白黒のピクチャのために実装され、ネイティブに管理されるピクチャには適しません。

## サポートされない機能

- 変数を要求するコマンドに 2D 配列の要素を渡すこと: これはエラーを返します。  
例: `Type(My2DArray{1}{1})`
- Mac OS 7 ポップアップメニューの擬似配列: ポップアップ変数が実数型で"pop"という名前がつけられているとき、変数の内容は"pop"+`String(pop)`でアクセスすることができました。
- 名前で変数にアクセスするためのパラグラフ演算子 §  
例えば: `a:= § ("vtoto")`  
この古い演算子はすでにコンパイラにより拒否されていました。  
例:  
`ValueOfA := § ("A")`  
は以下のように書き直します:  
`ValueOfA :=Get pointer("A")->`
- CLEAR VARIABLE** はクォーテーションマークに挟まれた変数名を引数にとることはできません。このコマンドはインタープリタおよびコンパイルモード両方で同じ動作をします。
- `arr42` にアクセスするための `arr{42}` や `arr{"42"}` のような擬似配列。
- タブに関連づけられた変数"toto"と、それぞれのタブの有効/無効を切り替えるブール配列"toto"。
- DDETools: この製品はサポートされません。代替は:  
Windows: VBA と **LAUNCH EXTERNAL PROCESS**  
Mac OS: Applescript と **LAUNCH EXTERNAL PROCESS**

## 動作の変更

### テーブルフォームとプロジェクトフォーム

テーブルに属さない新しいタイプのフォーム、プロジェクトフォームを作成できます。プロジェクトフォームは入力および出力フォームの両方に使用することができます。

### フォームのバックグラウンドで繰り返される小さなビットマップピクチャと

#### CPU

ピクチャ処理の変更により、バックグラウンドで繰り返される小さなビットマップピクチャをフォームで使用している場合、4D v11 の使用する CPU が著しく増えていることに気づくかもしれません。ピクチャが小さければ小さいほど、繰り返し数が多くなり、使用される CPU が多くなります。

この解決法は大きなピクチャを使用することです (最低 128 x 128)。これは多少より多くのメモリを必要としますが、CPU の処理時間を節約します。



## テーブルとフィールドの削除が可能に

4D v11 SQL ではテーブルとフィールドの削除が可能になりました。

削除されたテーブルのフォームはプロジェクトフォームとなり、エクスプローラのゴミ箱に移動されます。

削除されたテーブルとフィールドの番号は、新しくそれらが作成されたとき再利用されます。

廃止されたコマンド:	置き換え:
<b>Count tables</b>	<b>Get last table number</b>
<b>Count fields</b>	<b>Get last field number</b>

追加されたコマンド:

- Is table number valid**
- Is field number valid**

## トランザクション

**VALIDATE TRANSACTION** と **CANCEL TRANSACTION** は、カレントセレクションに影響を与えなくなりました。

以前のバージョンでは、トランザクション中でひとつでもレコードが作成されると、セレクションは0となりました。

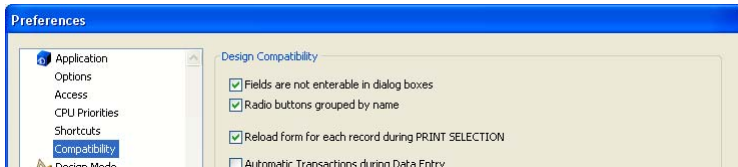
今バージョンからは、トランザクション中で作成されたレコードは (18,000,000 で始まる) 一時的な番号ではなく、標準の番号を使用します。この番号はトランザクションが受け入れられたとき永久化されます。トランザクションがキャンセルされると解放されます。

レコード番号に対して使用するコマンド (**SCAN INDEX**、**RELATE ONE SELECTION** 等) をトランザクション中で利用可能になります。

## ネストしたトランザクション

ネストしたトランザクションを有効にするには、環境設定で設定します。

4D 2004 ではネストしたトランザクションを使用できませんでした。トランザクション中にさらにトランザクションを開始しても、4D は二番目のトランザクションを無視していました。



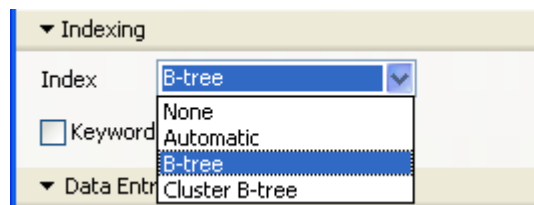
このオプションは変換されたデータベースにのみ表示されます。デフォルトでこのオプションはチェックされていません。

このオプションは4D v11 SQLのSQLエンジンによるトランザクションには影響しません。SQLエンジンのトランザクションは常にマルチトランザクションです。

## インデックス

リレーションがはられたフィールドに対するインデックスは、必須ではなくなります。

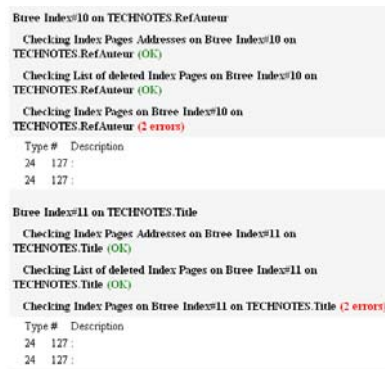
デフォルトで、インデックスはB-Treeに変換されます。



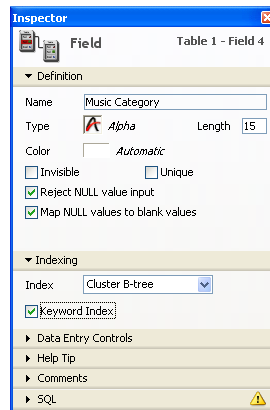
通常のテーブルに変換された以前のサブテーブルに追加されたフィールドには、クラスタ B-Tree インデックスが設定されます。

- 標準の B-tree: 多目的のインデックス。以前のバージョンの 4D で使用されていました。
- クラスタ B-tree: データ中に同じ値が繰り返し表れるときに有効なインデックス (例: ブールフィールドや、Mr. Mrs. Ms. で構成される敬称など)。
- 自動 (デフォルト): データに基づき 4D がどのインデックスを使用するか決定します。

これは Maintenance & Security Center でインデックスを検証した場合の例です:



- キーワードインデックス:

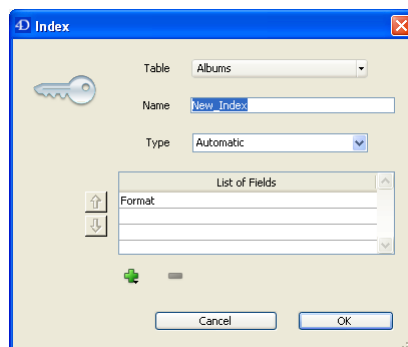


この新しいタイプのインデックスは、文字型またはテキストフィールドで利用できます。標準のインデックスと同時に利用できます。4D はコンテキストに応じて、より効率の良いインデックスを使用します。

現時点で、キーワードインデックスは半角スペースやピリオドなどで区切られた単語をインデックス化しません。日本語のような単語の切れ目がいまいな言語では、正しいインデックスを作成することはできません。ただし、日本語なども半角スペースで区切ることで、キーワードインデックスを作成することが可能です。

- 複合インデックス:

複合インデックスは、複数のフィールドから構成されるインデックスです。このインデックスは、(ストラクチャエディタのツールバーから呼び出せる) インデックスリストダイアログボックスでのみ作成できます。



## 新しい"キーワードを含む"演算子

この新しいタイプのインデックスのために、新しい演算子が用意されました: % は"キーワードを含む"に対応します。

単語 "easy" を含む検索:

**QUERY**([Products];[Products]Description%"easy")

"easy" で始まる単語を含む検索:

QUERY([Products];[Products]Description%"easy@")

## セット

削除されたレコードの参照を含むため、既にセットが有効でない状況で **USE SET** コマンドを使用すると、4D v11 SQL は以下のエラーを返します: -10503: "無効なレコード番号"

4D 2004 ではエラーは返されませんでした。

**Is in set:** このコマンドはテスト対象としてセレクションポインタを使用していました。セレクションポインタは、セレクション中のカレントレコードを指定します。ところで、セレクションポインタは時に実際にロードされているカレントレコードと異なるレコードを指すことがあります。例えば **PUSH RECORD** – セレクション (とカレントレコード) の変更 – **POP RECORD** の処理を行った場合に発生することがあります。この場合、テスト対象は (POP した) 現在のカレントレコードではなく、セレクションポインタが指す (前の) カレントレコードが対象でした。

これに対し **ADD TO SET** は常に実際のカレントレコードを対象としています。

今バージョンより、**Is in set** の動作は **ADD TO SET** と同様、ロードされているレコードが対象となります。

注: **CREATE SET** は、ロードされたレコードに関わらず、常にセレクションポインタを保持します。

## クリップボードとペーストボード

ペーストボードコマンドは、ペーストボード (以前のクリップボード) やドラッグ&ドロップデータを管理するために使用します。4D v11 SQL では実際二つのタイプの"ペーストボード"を使用しています: ひとつはカットまたはコピーされたデータで、もうひとつはドラッグおよびドロップされたデータです。

ドラッグ&ドロップデータのペーストボードには、[On Begin Drag Over](#)、[On Drag Over](#)、[On Drop](#) フォームイベントや、[On Drop](#) データベースメソッドのコンテキストのみでアクセスできます。

"クリップボード"テーマは"ペーストボード"に名称変更されました。

[No such data in clipboard](#) 定数は [No such data in pasteboard](#) に名称変更されました。

## メニュー

4D v11 SQL から、メニューは階層化されました。デザインモードで作成されていなくても、メニューやメニューバーをオンザフライで作成できます。メニューとメニューバーは同様に処理されます。最大文字数は 31 から 150 に変更されました。

追加された最後の項目を示すために、menuItem=-1 パラメタを使用できます。

以前の名称 (4D 2004.x)	新しい名称 (4D v11SQL)
<b>MENU BAR</b>	<b>SET MENU BAR</b>
<b>HIDE MENU BAR</b>	"ユーザインターフェース"テーマに移動
<b>SHOW MENU BAR</b>	"ユーザインターフェース"テーマに移動
<b>SET ABOUT</b>	"ユーザインターフェース"テーマに移動
<b>SET MENU ITEM KEY</b>	<b>SET MENU ITEM SHORTCUT</b>

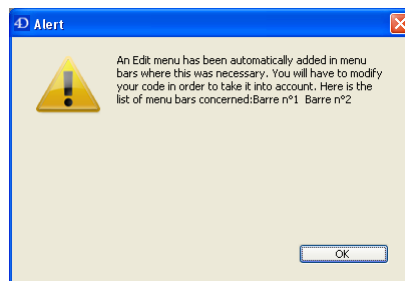
変更なし:

**Menu selected**  
**Count menus**  
**Count menu items**  
**DISABLE MENU ITEM**  
**ENABLE MENU ITEM**  
**APPEND MENU ITEM**  
**INSERT MENU ITEM**  
**DELETE MENU ITEM**

## 編集メニューの特別な機構

4D v11 SQL では、v6.8 オプションが使用できなくなりました。このオプションは変換されたデータベースで、システムによって管理される編集メニューを自動で追加する過去の機能の互換性のために使用されました。

v6.8 のオプションがチェックされたデータベースを変換すると、編集メニューが自動で追加され、警告ダイアログが表示されます:



警告: この結果、メニューを管理実行するコードの変更が必要になります。以前の機能においては、編集メニューはカウントされませんでした。しかし 4D v11 SQL では編集メニューも必ずカウントされます。

## 階層リスト

構造的な変更があり、より自由度が増しました。

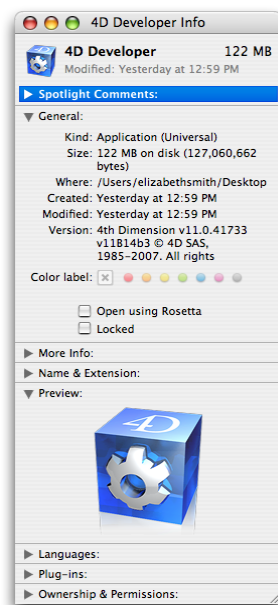
## トリガ

On Saving New Record トリガの動作に小さな変更があります: 2004 ではトリガで **Record number** コマンドを呼び出すと-3が返されました。4D v11 SQL では、このコマンドは新規に作成されるレコードに割り当てられるレコード番号が返されます。

## プラグイン

バージョン 2004 のプラグイン (4D 社およびサーボドパーティー製) は 4D v11 SQL と互換があります。

MacIntel: 4D Draw と 4D for OCI を除き、すべての 4D 社製 4D v11 SQL プラグインは MacIntel ネイティブです。4D Draw と 4D for OCI は Rosetta で使用しなければなりません。これを行うには、Finder 上で 4D Developer.app を選択し、"情報を見る"メニューを選択して情報を表示させ、"Rosetta を使って開く"にチェックを入れます。



## Plugin SDK

EX\_FIND\_PROCID と EX\_CALL\_BY\_PROCID はメソッド配列の添字を使用して動作していました。これらのエントリーポイントは以降、正数または負数の内部的な ID を使用して動作します。ヌル値は使用できません。

## バーチャルストラクチャ

バーチャルストラクチャは **SET TABLE TITLES** や **SET FIELD TITLES** で設定されたテーブル名やフィールド名を使用します。これらの名前は標準の 4D ダイアログボックス (クエリ、並び替え、クイックレポートなど) で使用されます。この動作は 4D v11 SQL で変更されました: 今バージョンからバーチャルストラクチャは、これらのダイアログボックスがランゲージコマンドから呼び出された場合にのみ適用されます。プラグインも常にバーチャルストラクチャにアクセスします:

	ストラクチャ	テーブルやフィールドの可視/不可視
デザインモード	実際のストラクチャ	可視
プラグイン	バーチャルストラクチャ	不可視
アプリケーションモード	バーチャルストラクチャ	不可視

## ピクチャ

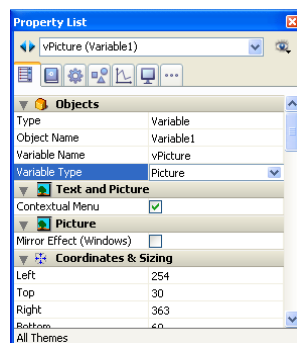
ピクチャはネイティブフォーマットで保存されます。

以前の **PICTURE TYPE LIST** コマンドは互換性のため保持されます。しかしこのコマンドは QuickTime を必要とし、4D がネイティブに管理するフォーマットを返すわけではありません。4D がネイティブで処理する画像のフォーマットを知るには **PICTURE CODEC LIST** を使用してください

## ピクチャ変数の型宣言

4D v11 SQL のインタープリタモードで、ピクチャ変数の内容が正しく表示されない場合、その変数が正しく型宣言されているか確認してください。これを行うには二つの方法があります:

- プロパティリストのポップアップメニューを使用してフォームレベルで型宣言を行う:



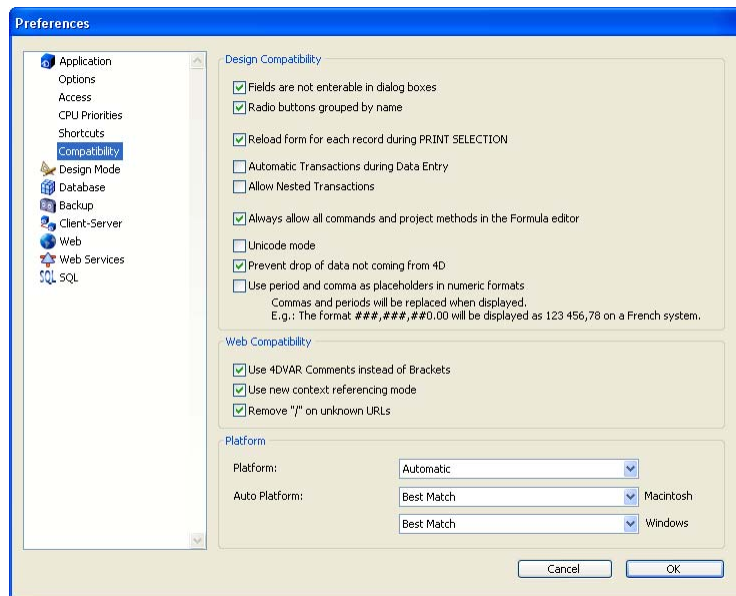
- またはコンパイラ宣言 **C\_PICTURE(vPict)** を使用する。この場合、フォームがロードされる前にコードが実行されていなければなりません。これは [On Load](#) フォームイベントより前ということです。

以前のバージョンの 4D では、一般的に変数の初期化はあいまいさが許されていました。フォーム上のピクチャ変数の表示は新しいネイティブのメカニズムで管理されます。

## ピクチャフィールドへの情報格納:

ピクチャの最後に、4D は X/Y 情報と表示モードを追加していました。この情報はもう必要ありません。この 6 バイトの情報は変換時に削除されます。もし何らかの理由で、ピクチャフィールドに Blob 等を保存していた場合、変換時に情報が一部失われてしまうことに注意してください。

## 環境設定の互換性:



## デザイン互換性:

4D 2004 で提供された互換性オプション:

- **ダイアログボックスでフィールドを入力不可にする:**  
(変換されたデータベースではチェックされ、新規データベースではチェックされていません。)
- **ラジオボタンを名前グループ化する:**  
ラジオボタンを関連づけるために、その変数名を同じ文字で始める必要はありません。単にグループ化するだけで関連づけられます。このオプションは変換されたデータベースでのみ表示され、デフォルトでチェックされています。v2004 で作成されたデータベースではこのあたり足胃機能を使用されています。
- **PRINT SELECTION 中、レコード毎にフォームをリロードする:**  
このオプションは変換されたデータベースで表示され、デフォルトでチェックされます。  
この機能によりすべてのオブジェクトパラメータが自動でリセットされ、開発者は、[On Printing Detail](#) フォ



ームイベントで言語を使用してそれらを変更できます。パフォーマンスを最適化させるため、4D 2004 でこの機能は削除されました。開発者は特定のパラメータを、フォームメソッドを使用してリセットしなければなりません。この機能はリストフォームの [On Display Detail](#) イベントと同じです。開発者は"PRINT SELECTION 中、レコード毎にフォームをリロードする"オプションにチェックして、以前のメカニズムを保持することができます。4D v11 で作成されたデータベースはこの新しい機能を使用します。

- データ入力時に自動トランザクションを使う:

このオプションはバージョン 2004 に変換された古いデータベースにのみ表示され、デフォルトでチェックされていません。この機能は、サブフォームを含む入力フォームを開いたときに、自動でトランザクションを開始するために使用されていました。このオプションは 4D にスイッチした 4D First ユーザのために用意されました。このオプションは 2003 の時点で非推奨でした。

- フォーミュラエディタで、常にコマンドやプロジェクトメソッドの利用を許可する:

現在フォーミュラエディタには、限られた 4D コマンドテーマのみが表示され、プロジェクトメソッドは表示されません。この制限を緩和するには、**SET ALLOWED METHODS** コマンドを使用するか、このオプションにチェックします。

## **Web 互換性:**

- ブラケットの代わりに 4DVAR コメントを使用する:

以前のバージョンの 4D では以下のシンタックスが使用されていました: [MyVar]。この記法は <!--4DVAR MyVar--> に置き換えられていました。

- 新しいコンテキスト参照モードを使用:

4D Web サーバが、ページの要素ごとにコンテキスト番号を送信しないようにします。

- 未知の URL から "/" を取り除く:

例えば以前のバージョンで、http://127.0.0.1/Contacts/Add の URL では\$1 に"Contacts/Add"が渡されました。4D 2004 では、すべての URL が"/"で始まります (このオプションがチェックされている場合を除く)。

- Keep-Alive 接続:

HTML ページが 10 の画像を含む場合、HTTP1.0 では 10+1 の接続がはられます。このオプションを使用するとひとつの接続で管理されます。

## **プラットフォーム**

環境設定ダイアログボックスで、このエリアは変換されたデータベースでのみ表示されます。以前のバージョンでは、データベースレベルやフォームレベル、およびオブジェクトレベルでアピアランスを強制することが可能でした (デフォルトは上位階層からの継承)。このメカニズムは互換性のため保持されています。

4D 2004 以降で作成されたデータベースは、プラットフォームに従い、4D が自動でインターフェースを選択します。フォームやオブジェクトはネイティブな仕様に従い表示されます。インターフェースはフォームやオブジェクトレベルでのみ設定可能で、選択可能なオプションは以下の通りです: フォームレベルではシステムまたは印刷、オブジェクトレベルではフォームから継承。

過去のバージョンの 4D から変換されたデータベースでは、追加のオプションが選択できます (自動、Mac OS 7、Windows 3.11、NT 3.51、Windows 95/98/2000、NT 4、Mac OS 9、Mac テーマ)。これらのオプションは 4D の過去のプラットフォームインターフェースサポートモードに対応しています。なおこれらのオプションの利用はお薦めしません。 **Get platform interface** と **SET PLATFORM INTERFACE** コマンドも同様であり、両コマンドとも互換性のために保持されていますが、変換されたデータベースでのみ動作します。これらのコマンドは 4D 2004 で作成されたデータベースでは無視され、プラットフォームインターフェースは自動で管理されます。

変換されたデータベースの環境設定で、この機能を有効にするために、システムオプションを選択することを強くお薦めします。その結果他の二つのポップアップメニューは選択不可となり、この新しいメカニズムが有効になります。このメカニズムを利用するためには、フォームのプラットフォームプロパティを変更する必要があることに注意してください。例えば、データベースから継承オプションを選択します。

## **新しい互換性オプション:**

### **外部からのドラッグアンドドロップを拒否**

アプリケーション/互換性ページ:

4D v11 SQL では、4D 外部のセレクションやオブジェクト、またはファイルなどのドラッグとドロップをサポートしています。例えばピクチャファイルなどです。この機能はデータベースコードでサポートされなければなりません。

過去のバージョンから変換された 4D では、もし既存のコードがこれに対応していない場合、この機能により予期しない動作になることがあります。このため、環境設定の新しいオプションを使用して、データベースへの外部オブジェクトのドロップを許可しないようにすることができます。

変換されたデータベースでは、デフォルトでチェックされています。

## トランザクションのネストを許可する

4D v11 SQL は無制限のネストされたトランザクションをサポートします。この新しい機能は変換されたデータベースではデフォルトで無効になっています。トランザクションは1レベルに制限されます。

変換されたデータベースで複数レベルのトランザクションを使用するには、このオプションをチェックしなければなりません。

注: SQL トランザクションは常にマルチレベルです。

## Unicode モード

このオプションは過去のバージョンから変換されたデータベースにのみ表示され、デフォルトではチェックされていません。

4D v11 SQL は以下の場所で Unicode を使用します:

- データベースエンジンと SQL サーバ
- 変数
- Web
- 多くのストラクチャオブジェクト: メニュー、リスト、ストラクチャ、ユーザ&グループ、ヘルプ Tips、その他

しかしながら現時点で、Unicode はフォームエディタやメソッドエディタでは使用されません。 - ローカライズには XLIFF を使用します。

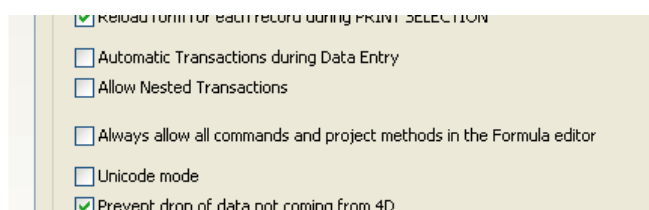
環境設定:

4D v11 では Unicode サポートが拡張され、多くのコマンドが書き直されました。

今バージョンより、データベースエンジン、ランゲージ、および (メニュー、リスト、フォームオブジェクト等) 多くのインターフェース要素で Unicode 文字を使用できます。

Unicode モードが有効になると、フォームオブジェクトの"キーボードレイアウト"プロパティは無視されません。

互換性のため、環境設定のオプションを使用して、4D v11 SQL を以前のモードで動作させることができます:



4D v11 SQL は UTF-16 をメインに、Web や書き出し/読み込みには (ほとんどのケースでデータサイズがコンパクトになるため) UTF-8 を使用します。Unicode では、1 から 127 までの値は ASCII 文字と一致します。

Unicode がサポートされたことにより、引数の変更や、新しい引数の追加、新しいコマンドの追加が行われています:

	非 Unicode	Unicode - UTF 16
文字フィールド	文字フィールド	文字フィールド
テキストフィールド	テキストフィールド	テキストフィールド
<b>C_TEXT</b>	テキスト型	テキスト型
<b>C_STRING</b>	文字変数型	テキスト型 文字長引数は無視
<b>ARRAY TEXT</b>	テキスト配列型	テキスト配列型
<b>ARRAY STRING</b>	文字配列型	テキスト配列型 文字長引数は無視
<b>Char</b>	<b>Char</b> (Ascii コード) 0-255	<b>Char</b> (UTF-16 値) 0-65535
<b>Ascii</b> は <b>Character code</b> に変更	<b>Character code</b> ASCII コードを返す	<b>Character code</b> Unicode コードを返す
<b>Length, String, Position, Delete string, Substring, Change string</b>  文字参照[[1]]		UTF-16 文字で動作  X 番目の UTF-16 文字を返します。X バイト目の文字ではありません。
<b>Mac to Win, Win to Mac, Mac to ISO, ISO to Mac</b>		動作しません

新しいコマンド: <b>CONVERT FROM TEXT</b> <b>Convert to text</b>	-	<b>CONVERT FROM TEXT</b> (4Dtext;charSet;convertedBLOB) <b>Convert to text</b> (blob;charSet) -> Text
新しい定数名と定数: <b>BLOB to text</b> <b>TEXT TO BLOB</b>	<a href="#">C String</a> <a href="#">Pascal string</a> <a href="#">Text with length</a> <a href="#">Text without length</a>	<a href="#">Mac C String</a> <a href="#">Mac Pascal string</a> <a href="#">Mac text with length</a> <a href="#">Mac text without length</a> <a href="#">UTF8 C String</a> <a href="#">UTF8 text with length</a> <a href="#">UTF8 text without length</a>

これらの新しいメカニズムは1バイトでコードされた文字では特段の影響を与えませんが、日本語等2バイト文字の処理を簡単にします。

**警告:** 日本の開発者向けの重要なお知らせ：

非 Unicode モード（互換モード）は、メソッド実行時に Unicode コードの文字を、日本語環境においては Shift\_JIS コードに変換してから、コマンドによる演算を行います。しかしながら以下のような制限があります：

- Unicode から Shift\_JIS に変換されるため、Shift\_JIS に存在しない文字はメソッド実行時に消失します。
- 文字の比較はモードにかかわらず、Unicode のルールを使用して行われるため、結果は以前のバージョンと異なることがあります。
- 上記と同じ理由により、非 Unicode モードで String テーマのコマンドが期待どおりに動作しないことがあります。

このことから、4D v11 SQL 利用時には、Unicode モードを使用することを強くお勧めいたします。非 Unicode モードをご利用いただいても、完全な互換性は提供されません。

## **警告: 4D 2004 で複数の言語をデータベースに格納している場合**

キーボードレイアウトプロパティを使用する等して、複数の言語 (特に日本語や中国語等の2バイト文字) をデータベースのフィールドに格納している場合は、以下の手順に従います：

4D 2004 -> 4D v11 SQL への変換時に、4D はデータを Unicode に変換します。このとき 4D は、現在のシステムのスクリプトを参照し、それによって元のデータの文字セットを決定します。システムの言語が日本語の場合は Shift\_JIS が使用されます。複数の言語が使用されている場合、情報を失わないためには元の文字セットを知ることが重要ですが、これを自動で行うことはできません。開発者が文字セットを指定する必要があります。この情報は、変換するデータファイルと同階層に"multilang.txt"という名前のテキストファイル置いて、その中で指定します。テキストファイルのエンコーディングは ANSI または Mac Roman でなければなりません (UTF-8 や Unicode は使用できません)。

ファイルの定義:

テーブル番号; フィールド番号; サブフィールド番号 (オプション); 言語コード

それぞれの行は改行 (CR または CRLF) で終了します。空行やスペースは無視されます。

このファイルは、「変換時に」データファイルと同階層になければなりません。

例:

開発者は以下の言語を指定したいとします:

ギリシャ語: Table3-Field3 と Table4-Subfield7-Field1,

ロシア語: Table3-Field4 と Table4-Subfield7-Field2,

"multilang.txt"ファイルには以下の情報を書きます:

3;3;1049

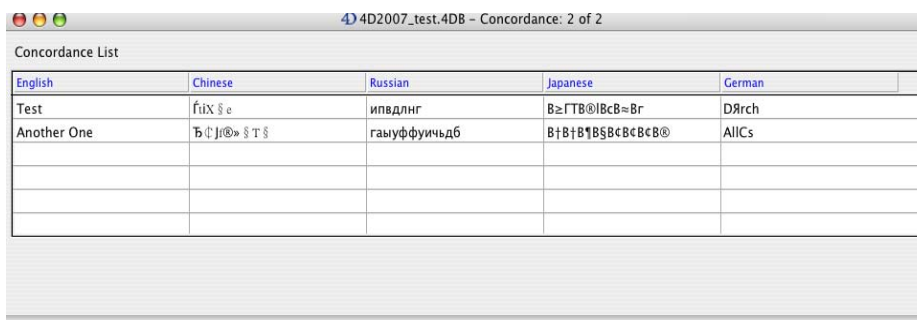
3;4;1032

4;7;1;1049

4;7;2;1032

言語コードは"keyboardmapping.xml"ファイルに書かれています (このドキュメントの appendix 1 を参照してください)。

"multilang.txt"ファイルを使用せずに v11 に変換されたデータベース:



English	Chinese	Russian	Japanese	German
Test	fiX § e	ипвдлнг	В≥ГТВ@ BcB=Br	DЯrch
Another One	Б†J @» § T §	гауффуичьдб	†††††B§BcBcBcB@	AllCs

変換時に"multilang.txt"ファイルをデータファイルと同階層に置いて変換したデータベース:



English	Chinese	Russian	Japanese	German
Test	差幾又	ипвдлнг	さけおんでゆ	Durch
Another One	哈搭型三卅	гауффуичьдб	ああえういおい	Alls

この手順に従えば、サポートされる言語を正しく変換することができます。

## 数値フォーマットにシステム設定を使用する:

このオプションは 4D v11 SQL で作成された新しいデータベースではチェックされています。この場合、4D は自動で数値表示フォーマットの","や"."を、オペレーティングシステムで定義された設定に置き換えます。

以前のバージョンでは、"###,##0.00"フォーマットは日本やアメリカのシステムで有効ですが、このフォーマットがフランスやスイスのシステムで表示されると結果が正しくないものとなりました。

今バージョンからは、環境設定でこのオプションをチェックし、システムチェックにコンマやピリオドを数値フォーマットのセパレータとして使用すれば、4D はこれらの文字をシステムで定義された文字に置き換えます。

**Num** コマンドに新しい引数を指定できるようになりました。この引数を使用して小数点に使用する区切り文字を指定できます。さらに、新しいコマンド **GET SYSTEM FORMAT** を使用して、地域のシステム設定を取得できます。

**警告:** 4D v11 SQL で新たに作成されたデータベースでは、オプションがチェックされている場合、データベースをマルチ OS 対応にするには、千の位の区切り文字を","に、小数点の区切り文字を"."にします。例えば"###,##0.00 €"のようにします。これにより、フォーマットはすべての OS で通常通り動作するようになります。

もし"### ##0.00 €"のような慣習的なフォーマットを引き続き使用したい場合は、環境設定のこのオプションのチェックを外します。そうでなければフォーマットを"###,##0.00 €" に変更します。

## サポートされない互換性オプション:

v3.x.x の Startup プロシージャ方式を使う: 今バージョンから、データベース開始時に実行させるコードは On Startup や On Server Startup データベースメソッドに置かれなければなりません。

v3.x.x の ファイルプロシージャ方式を使う: トリガに置き換えられました。

テキスト描画を 6.8 互換にする: (Quartz 互換でない、以前のスタイルのために使用されていました) – フォントにジャギーが見られる場合は、ビットマップフォントを使用していないか確認してください。

以前の編集メニューメカニズム: (v 6.8 オプション)

メニューエディタの v6.8 オプションは 4D v11 SQL ではサポートされません。このオプションは、変換されたデータベースで、システムにより管理される編集メニューを自動で追加するために使用されていました。今バージョンからは、編集メニューも他のメニューと同様に明示的に管理されなければなりません。もし編集メニューがメニューバーに存在しなければ、自動で追加されます (このドキュメント内の編集メニューの節を参照してください)。

# コマンド

## 新しいコマンド

新しい機能を管理するコマンドが数多く追加されました:

### Unicode サポート関連:

Get localized string

Match regex

CONVERT FROM TEXT

Convert to text

### SQL エンジンと SQL サーバ関連:

GET DATA SOURCE LIST

USE EXTERNAL DATABASE

USE INTERNAL DATABASE

Get current data source

QUERY BY SQL

Is field value Null

SET FIELD VALUE NULL

GET LAST SQL ERROR

START SQL SERVER

STOP SQL SERVER

### ネストしたトランザクション関連:

Transaction level

### クエリと並び替え関連:

新しいエンジンと検索タイプ、およびインデックスをサポートします:

CREATE INDEX

DELETE INDEX

SET QUERY AND LOCK

Find in field

キーワードによるクエリ



## プロジェクトフォーム関連:

NO DEFAULT TABLE

## ドラッグ&ドロップ関連:

4D はセレクション、オブジェクト、外部ファイルのドラッグ&ドロップを使用できます。

新しい [On Begin Drag Over](#) フォームイベント

新しいデータベースメソッド: [On Drop](#)

## ペーストボード関連:

新しい 4D シグネチャ:

SET FILE TO PASTEBOARD

Get file from pasteboard

GET PASTEBOARD DATA TYPE

Com.4d.text.native

Com.4d.text.utf16

Com.4d.text.rtf

Com.4d.picture.pict

Com.4d.picture.png

Com.4d.file.url

など

4 文字のタイプ (TEXT, PICT など) は互換性のため保持されます。

## メニュー関連:

Create menu

RELEASE MENU

GET MENU ITEMS

Get menu bar reference

GET MENU ITEM ICON

SET MENU ITEM ICON

Get menu item method

SET MENU ITEM METHOD

GET MENU ITEM PROPERTY

SET MENU ITEM PROPERTY

Get menu item modifiers

Dynamic pop up menu

SET MENU ITEM REFERENCE

Get menu item reference

Get selected menu item reference

### **階層リスト関連:**

SET LIST ITEM FONT

Get list item font

Find in list

SET LIST ITEM ICON

GET LIST ITEM ICON

SET LIST ITEM PARAMETER

GET LIST ITEM PARAMETER

LIST OF CHOICE LISTS

### **リストボックス関連:**

INSERT LISTBOX COLUMN FORMULA

SET LISTBOX TABLE SOURCE

GET LISTBOX TABLE SOURCE

### **プリント関連:**

OPEN PRINTING JOB

CLOSE PRINTING JOB

### **4D 環境関連:**

VERIFY DATA FILE

VERIFY CURRENT DATA FILE

Compact data file

OPEN SECURITY CENTER

Get current database localization

COMPONENT LIST

### **システム環境関連:**

Select RGB Color: カラー選択システムウインドウを表示

GET SYSTEM FORMAT: 地域のパラメタ値を取得

## ユーザインターフェース関連:

Tool bar height

## メソッド関連:

EXECUTE METHOD (パラメタ付き)

## 文字列、XLIFF、正規表現関連:

CONVERT FROM TEXT

Convert to text

Get localized string (XLIFF のみ)

Match regex

## ストラクチャ定義関連:

Is table number valid

Is field number valid

CREATE INDEX

DELETE INDEX

## Web 関連:

Validate Digest Web Password (Digest モードでの On Web Authentication データベースメソッド用)

## XML 関連:

DOM Find XML element by ID

DOM EXPORT TO PICTURE

## ピクチャ関連:

PICTURE CODEC LIST

TRANSFORM PICTURE (ピクチャ演算子の機能向上)

COMBINE PICTURES (同上)

CONVERT PICTURE

ネイティブピクチャフォーマット

## ツール関連:

Choose (criterion; value1 | ;value N |) Boolean | number

例:

VTitle:= **Choose** ([Contact]Masculine; " Mr "; " Mrs ")

**GET MACRO PARAMETER**

**SET MACRO PARAMETER**

## **シンタックスが変更されたコマンド:**

### **メニュー関連:**

メニュー関連のすべてのコマンドは、MenuRef タイプの文字列を menu パラメタに受け入れるようになり、また階層サブメニューが利用可能になりました。

**Menu selected** は階層メニューをサポートします。

### **階層リスト関連:**

コマンドはオブジェクト名を受け入れます。これはリストが複数表示されている場合に必須となります。

オブジェクト名や変数名とともに、階層リストで 4D の標準コマンド (**FONT, FONT STYLE, FONT SIZE, SET SCROLLBAR VISIBLE, SCROLL LINES, SET COLOR, SET RGB COLOR, SET FILTER, SET ENTERABLE**) が使用できます。

### **リストボックス関連:**

**INSERT LISTBOX COLUMN**

**INSERT LISTBOX ROW**

**DELETE LISTBOX ROW**

**GET LISTBOX ARRAYS**

リストボックスにおける **Focus object**:

フィールドが割り当てられた列: フィールドをさします

変数が割り当てられた列: 変数をさします

式が割り当てられた列: リストボックス変数をさします

**Displayed line number**: リストボックスで利用可能です

### **プリント関連:**

**PRINT SETTINGS**(dialType)

dialType: 0 すべて, 1: プリントフォーマット, 2: プリント

## コンポーネント関連:

**Structure file** (\*):(コンポーネントで利用された場合) ホストデータベースのストラクチャファイルが返されます。

**Get 4D folder**(folder;|\*) \*を指定すると、ホストデータベースフォルダが返されます。

**Is compiled mode** (\*)ホストデータベースの情報が返されます

## 4D 環境関連:

### SET DATABASE PARAMETER

#### Get database parameter

新しいセクタ: 41: Unicode mode 42: Temporary memory size 43: SQL Autocommit	変更されたセクタ: 17: Character set in Unicode mode IANA で定義された、使用されている文字セットモード 29: Web Log Recording 30: Client Web Log Recording 新しいログファイルフォーマット (CLF, DLF, ELF, WLF)	削除されたセクタ: 1 Seq Order Ratio 2 Seq Access Optimization 3 Seq Distinct Values Ratio 4 Index Compacting 5 Seq Query Select Ratio 26 Cache Writing Mode
--	--	---

### OPEN 4D PREFERENCES

4D 2004 キー:	変更または新しいキー
/Application/Compatibility/Structure Compatibility	/Application/Compatibility/Design Compatibility
/Design Mode/Method Editor/Styles for Syntax Elements	/Design Mode/Method Editor/Syntax Styles
/Database/Script Manager	/Database/International
/Client-Server/Publishing/Allow-Deny Table Configuration	/Client-Server/Publishing/Allow-Deny Configuration Table
	/Application/Access/General Settings
	/Design Mode/Structure/Automatic Form Creation
	/Moving
	/Moving/Default Actions during the Copy if Dependent Objects
	/Moving/Moving Dialog
	/Database/International/Right-to-left Languages
	/Database/International/Numeric Display Format

	/Web/Options/Options
	/Web/Log Format
	/Web/Log Format/Web Log Type
	/Web/Log Format/Web Log Token Selection
	/Web/Log Scheduler
	/Web/Log Scheduler/Backup Frequency for Web Log File
	/SQL
	/SQL/Configuration
	/SQL/Configuration/SQL Server Access

## システム環境関連:

### PLATFORM PROPERTIES

4D 2004 定数	新しい定数
Power Macintosh	Mac OS
Windows	Windows
Other G3 and above	PowerPC
Pentium	Intel Compatible
INTEL 386	
INTEL 486	
Macintosh 68K	
PowerPC 601	
PowerPC 603	
PowerPC 604	
PowerPC G3	

## データ入力とユーザインターフェース関連:

**DIALOG:** 新しい \* パラメタにより、フォームはカレントプロセス内で最後に開かれたウィンドウにロードされます。これにより新規プロセスを起動しなくてもフローティングパレットウィンドウを作成できます。

**Open window:** Mac OS でメタルルックを使用できます。

**SET FORMAT(\$MyThermometer; ";;;128" )**

サーモメータで、プログラムでバーバースョップモードを指定するために使用します (フラグ 128)。

**SET FORMAT(\$MyButton ;:#MyPictures/MyPicture.jpg)**

3D ボタンでロードするピクチャを参照するために使用します。上記の場合、"Resources"フォルダ内の MyPictures サブフォルダにある MyPicture.jpg がロードされます。

## GOTO AREA(\* ; "" )

オブジェクトのフォーカスが外れます。

## メソッド関連:

オブジェクトメソッドで実行された **Current method name** は、"[Table1].Form.var"を返していましたが、今バージョンからは"[Table1]Form.var"を返します ( "."が削除されます)。

## 文字列関連:

**Num** は小数点区切り文字の指定を受け入れるようになります。デフォルトで、オペレーティングシステムが指定する小数点区切り文字が使用されます。この引数を渡すと、このコマンドはシステムの指定する区切り文字を考慮しなくなります。ひとつ以上の文字を渡すことができます。

**String** は引数に文字またはブール ("True" "False") を受け入れるようになります。

## Lowercase, Uppercase

新しい \* パラメタを指定するとアクセントが保持されます

**Lowercase** (string! ;\*)

**Uppercase** (string! ;\*)

バージョン 2004.x では

**Uppercase** ("è") -> E : アクセントが失われる

**Lowercase** ("È") -> è : アクセントは保持される

4D v11 では

**Uppercase** ("é") -> E : アクセントが失われる (バージョン 2004 と同様)

**Lowercase** ("È") -> e : アクセントが失われる

**Uppercase** ("é";\*) -> È : アクセントは保持される

**Lowercase** ("È";\*) -> è : アクセントは保持される (バージョン 2004 と同様)

例:

**Lowercase** ("ÉÚÂÆ éABC") -> euæ eabc アクセントが失われ、すべて小文字になる

**Lowercase** ("ÉÚÂÆ éABC";\*) -> éúâæ éabc アクセントが保持され、すべて小文字になる

**Uppercase** ("éà") -> EA : アクセントが失われる

**Uppercase** ("éà";\*) -> ÉÀ : アクセントは保持される

## Position

新しい引数 start と\*が追加されました。\*引数は検索を、発音区別符号を考慮して行う場合に使用します。

**Position** 関数は書き直され、ICU (International Components for Unicode) ライブラリを使用します。この

ため若干の動作変更があります。今バージョンより **Position** は Unicode で無視される文字 (ASCII NULL や Char(1)) を考慮に入れなくなります。

警告: 4D 2004 と同じ動作をさせるためには、新しいパラメタ\*を指定することを忘れないでください:

\$p:=**Position(Char(0);vText)** \$p -> 1

0 を得るには、\*引数を指定します

\$p:=**Position(Char(0);vText;\*)** \$p -> 0

注: Unicode では、以下の文字は予約されており、テキストに含まれてはいけません:

0

65534 (FFFE)

65535 (FFFF)

## ストラクチャ定義関連:

**Field** (サブフィールドポインタの場合、サブフィールドの番号を返します)

**Field name** (サブフィールドポインタの場合、サブフィールドの名前を返します)

**SET INDEX** (新しいインデックスタイプを考慮します)

## Web 関連:

**SET HTML ROOT** (今バージョンから、このコマンドは環境設定を考慮に入れません)

## XML 関連:

**DOM Find XML element**

**DOM Parse XML source**

**DOM Parse XML variable**

**SAX ADD XML ELEMENT VALUE**

**APPLY XSLT TRANSFORMATION**

## システムドキュメント関連:

**Select folder:** 新しい defaultPath 引数



**Open document**(document|; fileType|; mode|)

**Create document**

**Append document**

これらの三つのコマンドの fileType パラメタが変更されました。セミコロンで区切られたドキュメントタイプリストを渡すことができるようになりました。

## **BLOB 関連:**

**TEXT TO BLOB**

**BLOB to text**

## **リソース関連:**

**Get indexed string**

**STRING LIST TO ARRAY**

XLIFF のサポート

## **通信関連:**

**SEND PACKET** (BLOB をサポート)

**RECEIVE PACKET** (BLOB をサポート)

**USE CHARACTER SET** (Unicode で、IANA 文字セット名をサポート)

## **Web サービス関連:**

**CALL WEB SERVICE** (\*を指定することで Keep-alive が利用可能)

## **プロセス関連:**

新しい定数:

<a href="#">Web server Process</a>	Longint	-13
<a href="#">Execute on Client Process</a>	Longint	-14
<a href="#">4D Server Process</a>	Longint	-15
<a href="#">On Quit Process</a>	Longint	-16
<a href="#">Method editor macro Process</a>	Longint	-17
<a href="#">Internal 4D Server Process</a>	Longint	-18
<a href="#">Backup Process</a>	Longint	-19
<a href="#">Log File Process</a>	Longint	-20
<a href="#">Restore Process</a>	Longint	-21
<a href="#">MSC Process</a>	Longint	-22

<a href="#">Timer Process</a>	Longint	-23
<a href="#">SQL Method Execution Process</a>	Longint	-24
<a href="#">Server Controller Process</a>	Longint	-25
<a href="#">Monitor Process</a>	Longint	-26

## **名称が変更されたコマンド:**

### **EXECUTE**

Find index key

Ascii

GET PLUGIN LIST

USE ASCII MAP

### **EXECUTE FORMULA**

Find in field

インデックスなしのフィールドでも動作します。またワイルドカードを使用できます (ただしこの場合シーケンシャルモードになります)。

Character code

PLUGIN LIST

USE CHARACTER SET

## **ペーストボード関連:**

Test clipboard

GET CLIPBOARD

CLEAR CLIPBOARD

APPEND TO CLIPBOARD

SET PICTURE TO CLIPBOARD

GET PICTURE FROM CLIPBOARD

SET TEXT TO CLIPBOARD

Get text from clipboard

Test pasteboard

GET PASTEBOARD DATA

CLEAR PASTEBOARD

APPEND DATA TO PASTEBOARD

SET PICTURE TO PASTEBOARD

GET PICTURE FROM PASTEBOARD

SET TEXT TO PASTEBOARD

Get text from pasteboard

## **メニュー関連:**

MENU BAR

SET MENU ITEM KEY

SET MENU BAR

SET MENU ITEM SHORTCUT

## **階層リスト関連:**

INSERT LIST ITEM

DELETE LIST ITEM

INSERT IN LIST

DELETE FROM LIST

## **ストラクチャ定義関連:**

Count tables

Count fields

Get last table number

Get last field number

## 配列関連:

DELETE ELEMENT  
INSERT ELEMENT

DELETE FROM ARRAY  
INSERT IN ARRAY

## 動作が変更されたコマンド:

**ALL RECORDS** no longer accepts an unpointed field as parameter.

### SET HTML ROOT(rootFolder)

このコマンドは非コンテキストモードでも動作するようになります。このコマンドはカレント Web セッションにおけるすべての Web プロセスのルートフォルダを変更します。環境設定で指定されるデフォルト HTML ルートのパス名は変更されません。rootFolder パラメタには以下の形式の文字列を指定できます:

Mac OS        Disk:Applications:myserv:folder

Windows      C:\Applications\myserv\folder

以前のシンタックス (相対 "URL" タイプ) も使用できます。

**GET WEB FORM VARIABLES** で使用する配列は事前に宣言されていなければなりません。

### Create document

4D v11 SQL では、Document 変数に完全なパス名が返されます。4D 2004 では **Create document** で指定された名前が返されました。

例:

```
$Ref:= Create document ("MyGreatDocument")
```

Document には以下の値が返されます

v2004        -> MyGreatDocument

v11 SQL     -> C:\MyFolder\MyDatabase\MyGreatDocument

印刷時のブレークレベル管理:

**ACCUMULATE** と **BREAK LEVEL**: コンパイルモードと同様、インタプリタモードでもこれらのコマンドの使用が必須となります。

**ACCUMULATE** はインタプロセス変数を使用できなくなります。

テキストフィールドと変数は、最初の 80 文字のみが考慮されました。4D v11 SQL ではすべてのテキストが考慮されます。

**CLEAR VARIABLE**: このコマンドは今バージョンより、インタプリタモードとコンパイルモードで同じ動作をします。両方のモードで、変数をデフォルト値に初期化します。

4D 2004: このコマンドは、インタプリタモードではメモリから変数を消去し、コンパイルモードでは値を

初期化しました。

**CLEAR VARIABLE**("toto") シンタックスは許可されません。

**Field** と **Field name**: フィールドへのポインタを渡すと、これらのコマンドは、ソースフィールドの代わりにサブテーブルのフィールドの番号や名前を返します (すでに 4D 2004 でこのように動作していましたが、ドキュメント化されていませんでした)。

**Selected record number**: **PRINT RECORD** のフレームワークでは 0 が返されました。4D v11 SQL では、実際の位置番号が返されます。

**SAVE SET**: 新しいフォーマットが使用されます。4D v11 SQL で保存されたセットを以前のバージョンで読み込むことはできません。しかし **LOAD SET** は以前のフォーマットを読み込むことができます。

**SEND RECORD**: 新しいフォーマットが使用されます。4D v11 で送信されたレコードを以前のバージョンで読み込むことはできません。しかし **RECEIVE RECORD** は、サブテーブルを含まない限り以前のフォーマットを読み込むことができます。(サブテーブルはサポートされません。)

**QUIT 4D** は非同期モードで動作します。

**DELAY PROCESS** はアプリケーションプロセスに対しても有効です。

**SELECTION TO ARRAY** は、リレートされたテーブルのセレクションを変更しなくなります。またリレートされたテーブルのカレントレコードをアンロードしません。

**ARRAY TO SELECTION** はテーブルに属さないフィールドを無視します。配列サイズが同じでない場合、エラーを返します。

**Self**: オブジェクトメソッドから呼ばれたプロジェクトメソッド内で使用される **Self** ポインタは、バージョン 2004 ではコンパイルモードではオブジェクトへのポインタを返しましたが、インタプリタモードでは **Nil** を返しました。4D v11 では両モードでオブジェクトへのポインタを返します。

文字位置参照のオーバーフロー [[i]]: エラーメッセージ (4D 2004 では表示されませんでした)。

## QUERY

[Table1]	[Table2]	[Table3]
Field1->	Field 1	
	Field2 ->	Field 1

**QUERY** ([Table3];[Table1] Field 1=1)

4D v2004 -> [Table3]のレコードは検索されません

4D v11 SQL -> [Table3]のレコードが検索されます

カレントレコードのアンロードとリロード:

**GOTO RECORD:** カレントレコードは READ ONLY / READ WRITE モードが変更された場合にのみ、リロードされます。**QUERY** 後の最初のレコードのロードも同様です。

**RELATE ONE SELECTION:** 4D 2004 はカレントレコードをアンロードしました。4D v11 SQL ではアンロードしません。

**Drop position:** 今バージョンより変数、テキストフィールド、およびコンボボックスでも動作します。

プロセス変数へのポインタ

インタープリタモードとコンパイルモードで動作が同じになります。

二次元配列関連の変更:

- 変数を期待するコマンドの引数に二次元配列を渡すと、4D はエラーを生成します。以前のバージョンでは、エラーは生成されませんでした。二番目の添字は無視されていました。
- Type 関数が二次元配列の"行"に適用されると、二次元配列の実際の型を返すようになります。"Array 2D" タイプではなくなります。
- 文字型の二次元配列: インタープリタモードで、文字長は変更できなくなります。

## **第一引数にテーブルが必須となるコマンド:**

**APPLY TO SELECTION**

**QUERY BY FORMULA**

**QUERY SELECTION BY FORMULA**

**EDIT FORMULA**

## **テーマが変更されたコマンド:**

“ユーザインターフェース” テーマ

**HIDE MENU BAR**

**SHOW MENU BAR**

**SET ABOUT**

**SHOW TOOL BAR**

**HIDE TOOL BAR**

“4D 環境” テーマ

SET DATABASE PARAMETER

Get database parameter

BUILD APPLICATION

PLUGIN LIST

“ツール” テーマ

LAUNCH EXTERNAL PROCESS

SET ENVIRONMENT VARIABLE

“ユーザ&グループ” テーマ

Is license available

“フォーミュラ” テーマ

EXECUTE FORMULA

## **最適化されたコマンド:**

4D v11 SQL で多くのコマンドが最適化されました。特に:

QUERY BY FORMULA

QUERY SELECTION BY FORMULA

QUERY WITH ARRAY

## **廃止されたコマンド:**

	以下に置換え
SEARCH BY INDEX	QUERY
SORT BY INDEX	ORDER BY
SAVE OLD RELATED ONE	SAVE RELATED ONE
Get component resource ID	-
DATA SEGMENT LIST	-
ADD DATA SEGMENT	-
Version type	-

## **実行しても意味のないコマンド:**

**REDRAW LIST** (hierarchical list)

## **無効なシンタックス:**

- **If (Record number**([Table])=-3)

このシンタックスは 4D v6.5 より推奨されていませんでした。On saving new record トリガで、この関数は -3 を返さず、レコード番号を返します。

このコードは以下に置き換えなければなりません:

**If (Is new record**([Table]))

フォームメソッドで、**If(Before)...** (**During**)... (**After**)のシンタックスは、フォームイベントのテストに置き換えなければなりません: **If(Form event=**On Load**)** 等

- サブリストを削除するために **CLEAR LIST** (list;\*)を使用している場合、**DELETE FROM LIST** コマンドに置き換えるべきです (**DELETE LIST ITEM** コマンドの新しい名前)。
- **\$Structure:=Structure file**  
**\$Resources:=Open resource file** (\$Structure)

## **サブレコードコマンド:**

v11 に変換後、サブテーブルは標準のテーブルになります。しかし自動で付加された特別なりレシーョンを削除しない限り、サブレコードコマンドは引き続き動作します。

このリレシーョンを削除したあとは、テーブル用のコマンドを使用しなければなりません。

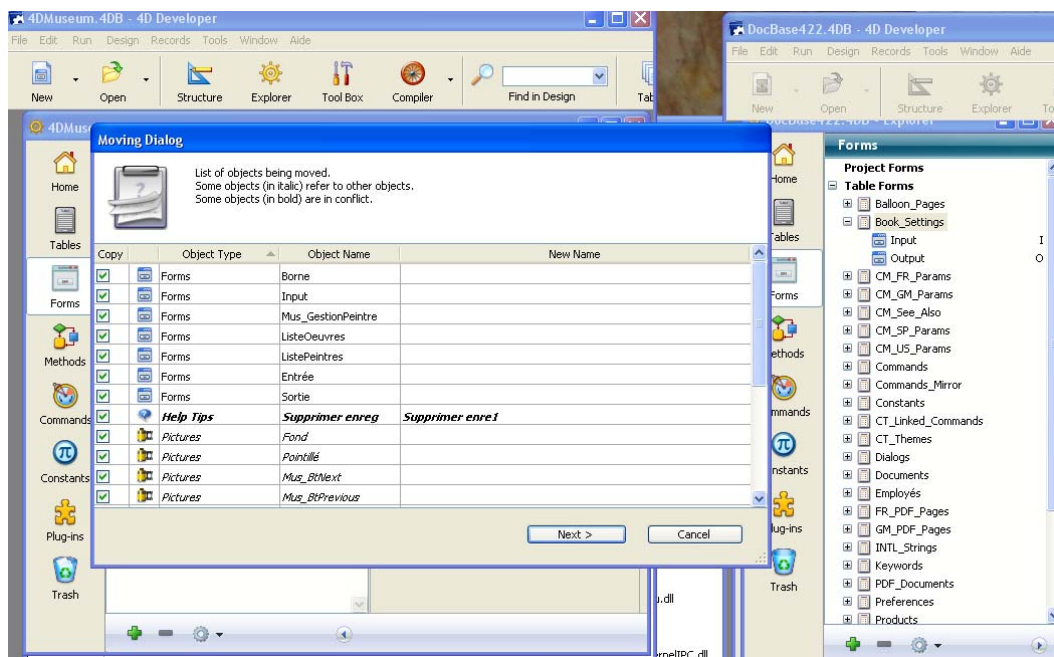
**警告:** **DUPLICATE RECORD** や **SEND RECORD**、**RECEIVE RECORD** はサブテーブルを無視します。

# 統合

## 4D Insider のオブジェクト移動と検索・置換:

データベース間でのオブジェクト移動

- 選択リスト
- スタイルシート
- フォーマットとフィルタ
- ピクチャライブラリのピクチャ
- メニューとメニューバー
- ヘルプ Tip
- プロジェクトフォームとそのフォームメソッド
- テーブルフォームとそのフォームメソッド
- プロジェクトメソッド
- フォルダ/サブフォルダ
- テーブルとそのフィールドおよびトリガ
- すべてのフォームオブジェクト (オブジェクトメソッド - 4D 2004)



名前の衝突がある場合は赤で表示されます。

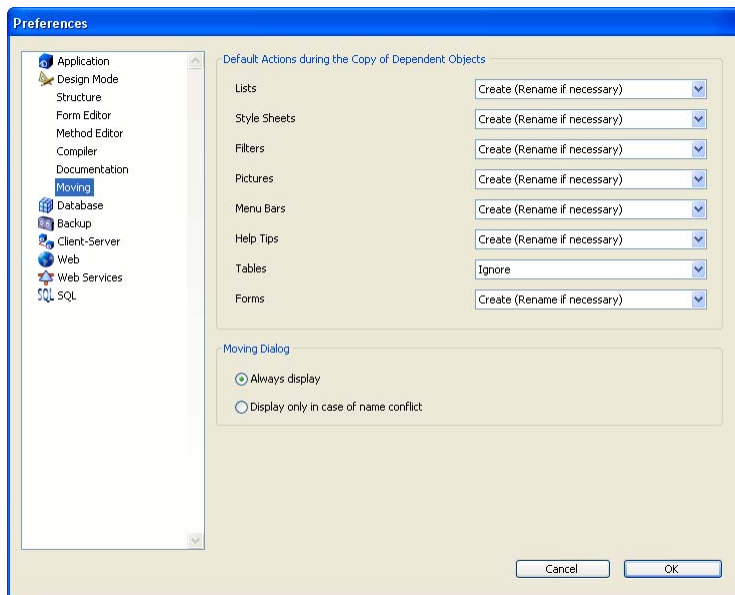
依存オブジェクトはイタリックで表示されます。

メニューにおいて、プロジェクトメソッドへの参照は保持されます。フォームの継承も保持されます。アクセス権は保持されません。



すべてを選択または選択解除するには、ALT+クリック(Win) OPTION +クリック (Mac) します。  
他のオブジェクトを使用: 依存オブジェクトにのみ存在します。

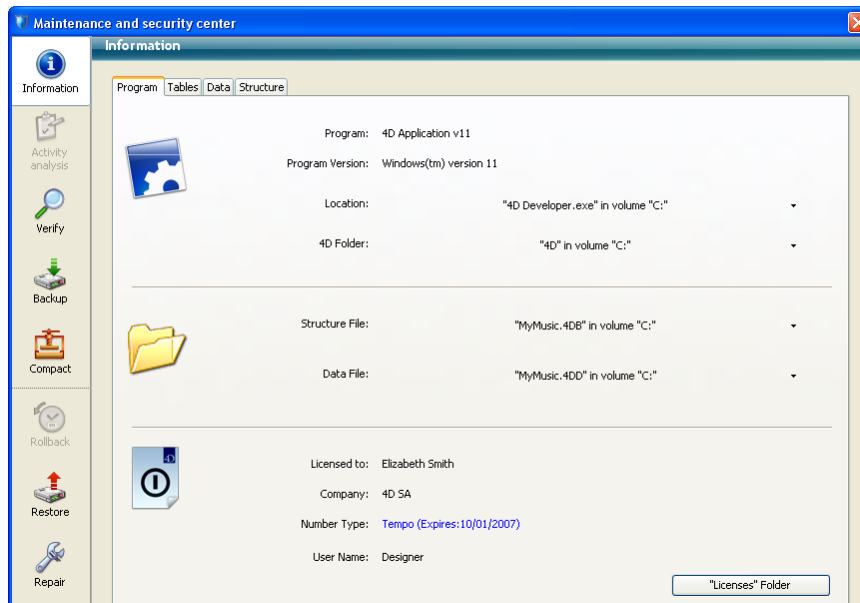
環境設定で移動オプションを設定できます:



## 4D Tools は MSC に

Maintenance & Security Center (MSC) が 4D Tools の代わりに実装されました。MSC は 4D に統合され、  
ツールバーからアクセスできます:





## 4D Pack

一部の 4D Pack コマンドの削除に関連する問題から、移行を簡単にするために、v11 で 4D Pack 2004 をロードすることができます。しかしながら、v11 で 4D Pack 2004 を使用することはお薦めしません。v11 での 4D Pack 2004 の利用はサポートできません。

### 廃止された 4D Pack コマンド:

既存のデータベース変換後、以下のコマンドは置き換えられなければなりません:

4D Pack コマンド:

以下に置き換え:

**AP PICT DRAGGER**

4D のドラッグ&ドロップ

**AP PICT UPDATER**

4D Pack の外部エリア %AP Pict displayer をピクチャフィールドに置き換えます

**AP Read Picture File**

**READ PICTURE FILE**

**AP Save BMP 8 bits, AP Save GIF**

**CONVERT PICTURE** および **WRITE PICTURE FILE**

**AP Read Picture BLOB**

ピクチャはネイティブフォーマットで保存されるようになったため、このコマンドは何も行いません。または **BLOB TO PICTURE** を使用

**AP PICT DRAGGER, AP Rect dragger**

MouseDown, MouseX および MouseY システム変数

**AP SET PICT MODE**

**SET FORMAT** を使用

**AP Select document**

**Select document** を使用

**AP Set palette**

**Select RGB color** を使用

**AP ShellExecute**

**LAUNCH EXTERNAL PROCESS** を使用

**AP Sublaunch**

**LAUNCH EXTERNAL PROCESS** を使用

4D 2004 ですすでに廃止されたコマンド:

**AP PrintDefault**

**AP Blob to print setting** および **AP Print settings to Blob** を使用

**AP Text to PrintRec**

**AP PrintRec to text**

**AP PrValidate**

**AP Toolbar installed**

**Tool bar height** を使用(インストールされていない場合 0 を返す)

**AP SET WEB FILTERS**

**SET DATABASE PARAMETER** を使用

**AP Add table and fields**

4D v11 SQL の新しい SQL 機能を使用

例:

```
CREATE TABLE Client
```

```
(Last name char(50),
```

```
First name char(50),
```

```
Address char(50),
```

```
City char(50),
```

```
Country char(25),
```

```
Birthdate date)
```

**AP Create relation**

現時点で代替の機能はありません (将来のバージョンで実装できよう検討中です)

## コマンドは存在するが使用を推奨しないもの:

**AP Save BMP 8 bits**

**AP Get picture type**

**AP GET PARAM**

多くのパラメタが無効となっています。

**AP SET PARAM**

**SET DATABASE PARAMETER** で置き換えてください。

**AP AVAILABLE MEMORY**

警告: 値は KB 単位となりました。

**AP FCLOSE**

**SET CHANNEL** や **SEND PACKET**、

**AP fopen**

**RECEIVE PACKET** で置き換え

**AP FPRINT**

**AP fread**

**AP CLOSE HELP**

Windows XP で動作しません。VISTA 非互換です。

**AP HELP INDEX**

**AP HELP ON HELP**

**AP HELP ON KEY**

## 向上したコマンド:

### AP Create method

propertiesArray に二つの値を追加できます:

- propertiesArray|5|に 1 を渡すと、メソッドはコンポーネントとホストデータベースで共有されます。
- propertiesArray|6|に 1 を渡すと、メソッドは SQL のコンテキストで利用可能です。

## 変更のないコマンド:

### AP Get templates

### AP Rect dragger

### AP SET CLOCK

### AP Timestamp to GMT

### AP FULL SCREEN

### AP NORMAL SCREEN

### AP BLOB to print settings

### AP Print settings to BLOB

### AP Get tips state

### AP SET TIPS STATE

### AP Get field infos

### AP Get table inf

## 新しいコマンド

### AP Get File MD5 Digest (filePath;digest;fork) -> Error

引数	タイプ		説明
filePath	テキスト	->	ファイルへの完全なパス名
Digest	倍長整数	<-	ファイルの MD5 ダイジェスト
fork	倍長整数	->	0: データフォーク 1: リソースフォーク

#### 説明

AP Get File MD5 Digest コマンドはドキュメントの MD5 ダイジェスト値を返します。

### C\_TEXT(\$document)

### C\_LONGING(<>digest;\$fork)

\$fork:=0

\$document:=**Select document**("" ; "" ; "" ;0)

\$error:= **AP Get File MD5 Digest** (\$document;<>digest;\$fork)

## ショートカット

データベースを開く際に、Alt (Windows) または Option (MacOS) を押下すると、データファイルを作成したり選択したりできます。

## デザインモード

プリント: CTRL/CMD P

用紙設定: CTRL/CMD SHIFT P

フォームを開く: CTRL/CMD L

エクスプローラを開く: CTRL/CMD SHIFT E

ツールボックスを開く: CTRL/CMD SHIFT T

コンパイラ: CTRL/CMD \*

デザインモードを検索: CTRL/CMD SHIFT F

環境設定: CTRL/CMD /

アプリケーションテスト: CTRL/CMD I

メソッド実行: CTRL/CMD R

インタープリタモードで実行: CTRL/CMD ALT I

コンパイルモードで実行: CTRL/CMD SHIFT I

ウィンドウ最小化: CTRL/CMD M

すべてのウィンドウを最小化: CTRL/CMD ALT SHIFT M

### ストラクチャ:

すべてを選択 (テーブルまたはフィールド): CTRL/CMD A

連続または非連続セレクション (テーブルまたはフィールド): CLICK SHIFT CLICK または CLICK

CTRL/CMD CLICK

次のテーブル/前のテーブル: Tab および Shift Tab

フィールドの作成: キャリッジリターン

### メソッド:

メソッド実行: CTRL/CMD K

メソッドエディタ上でメソッドを開く: ALT DOUBLE CLICK

行番号の表示/非表示: CTRL/CMD ALT SHIFT N

次のエラー: CTRL/CMD =

前のエラー: CTRL/CMD ALT =

ブロックを選択: CTRL/CMD B

コメント/アンコメント: CTRL/CMD /

式の入れ替え: CTRL/CMD ALT T

フォーム:

前面へ: CTRL/CMD F

背面へ: CTRL/CMD SHIFT B

グループ化: CTRL/CMD G

グループ解除: CTRL/CMD SHIFT G

オブジェクトのヘルプ Tip: オブジェクト上で CTRL/CMD SHIFT

## **レコード**

テーブルリスト: CTRL/CMD SHIFT U

カレントテーブル表示: CTRL/CMD U

リスト中に新規レコード: CTRL/CMD SHIFT N

新規レコード: CTRL/CMD ALT N

すべて表示: CTRL/CMD +

サブセット表示: CTRL/CMD -

並び替え: CTRL/CMD SHIFT Y

フォーミュラで更新: CTRL/CMD SHIFT R

検索: CTRL/CMD Y

データバッファをフラッシュ: CTRL/CMD ALT SHIFT S

## Appendix 1: ランゲージコード:

これらのランゲージコードは keyboardmapping.xml ファイルに記載されています。

DC\_AFRIKAANS = 1078  
DC\_ALBANIAN = 1052  
DC\_ARABIC\_SAUDI\_ARABIA = 1025  
DC\_ARABIC\_IRAQ = 2049  
DC\_ARABIC\_EGYPT = 3073  
DC\_ARABIC\_LIBYA = 4097  
DC\_ARABIC\_ALGERIA = 5121  
DC\_ARABIC\_MOROCCO = 6145  
DC\_ARABIC\_TUNISIA = 7169  
DC\_ARABIC\_OMAN = 8193  
DC\_ARABIC\_YEMEN = 9217  
DC\_ARABIC\_SYRIA = 10241  
DC\_ARABIC\_JORDAN = 11265  
DC\_ARABIC\_LEBANON = 12289  
DC\_ARABIC\_KUWAIT = 13313  
DC\_ARABIC\_UAE = 14337  
DC\_ARABIC\_BAHRAIN = 15361  
DC\_ARABIC\_QATAR = 16385  
DC\_BASQUE = 1069  
DC\_BELARUSIAN = 1059  
DC\_BULGARIAN = 1026  
DC\_CATALAN = 1027  
DC\_CHINESE\_TRADITIONAL = 1028  
DC\_CHINESE\_SIMPLIFIED = 2052  
DC\_CHINESE\_HONGKONG = 3076  
DC\_CHINESE\_SINGAPORE = 4100  
DC\_CROATIAN = 1050  
DC\_CZECH = 1029  
DC\_DANISH = 1030  
DC\_DUTCH = 1043  
DC\_DUTCH\_BELGIAN = 2067  
DC\_ENGLISH\_US = 1033  
DC\_ENGLISH\_UK = 2057  
DC\_ENGLISH\_AUSTRALIA = 3081

DC\_ENGLISH\_CANADA = 4105  
DC\_ENGLISH\_NEWZEALAND = 5129  
DC\_ENGLISH\_EIRE = 6153  
DC\_ENGLISH\_SOUTH\_AFRICA = 7177  
DC\_ENGLISH\_JAMAICA = 8201  
DC\_ENGLISH\_CARIBBEAN = 9225  
DC\_ENGLISH\_BELIZE = 10249  
DC\_ENGLISH\_TRINIDAD = 11273  
DC\_ESTONIAN = 1061  
DC\_FAEROESE = 1080  
DC\_FARSI = 1065  
DC\_FINNISH = 1035  
DC\_FRENCH = 1036  
DC\_FRENCH\_BELGIAN = 2060  
DC\_FRENCH\_CANADIAN = 3084  
DC\_FRENCH\_SWISS = 4108  
DC\_FRENCH\_LUXEMBOURG = 5132  
DC\_GERMAN = 1031  
DC\_GERMAN\_SWISS = 2055  
DC\_GERMAN\_AUSTRIAN = 3079  
DC\_GERMAN\_LUXEMBOURG = 4103  
DC\_GERMAN\_LIECHTENSTEIN = 5127  
DC\_GREEK = 1032  
DC\_HEBREW = 1037  
DC\_HUNGARIAN = 1038  
DC\_ICELANDIC = 1039  
DC\_INDONESIAN = 1057  
DC\_ITALIAN = 1040  
DC\_ITALIAN\_SWISS = 2064  
DC\_JAPANESE = 1041  
DC\_KOREAN\_WANSUNG = 1042  
DC\_KOREAN\_JOHAB = 2066  
DC\_LATVIAN = 1062  
DC\_LITHUANIAN = 1063  
DC\_NORWEGIAN = 1044  
DC\_NORWEGIAN\_NYNORSK = 2068  
DC\_POLISH = 1045  
DC\_PORTUGUESE = 2070



DC\_PORTUGUESE\_BRAZILIAN = 1046  
DC\_ROMANIAN = 1048  
DC\_RUSSIAN = 1049  
DC\_SERBIAN\_LATIN = 2074  
DC\_SERBIAN\_CYRILLIC = 3098  
DC\_SLOVAK = 1051  
DC\_SLOVENIAN = 1060  
DC\_SPANISH\_CASTILLAN = 1034  
DC\_SPANISH\_MEXICAN = 2058  
DC\_SPANISH\_MODERN = 3082  
DC\_SPANISH\_GUATEMALA = 4106  
DC\_SPANISH\_COSTA\_RICA = 5130  
DC\_SPANISH\_PANAMA = 6154  
DC\_SPANISH\_DOMINICAN\_REPUBLIC = 7178  
DC\_SPANISH\_VENEZUELA = 8202  
DC\_SPANISH\_COLOMBIA = 9226  
DC\_SPANISH\_PERU = 10250  
DC\_SPANISH\_ARGENTINA = 11274  
DC\_SPANISH\_ECUADOR = 12298  
DC\_SPANISH\_CHILE = 13322  
DC\_SPANISH\_URUGUAY = 14346  
DC\_SPANISH\_PARAGUAY = 15370  
DC\_SPANISH\_BOLIVIA = 16394  
DC\_SPANISH\_EL\_SALVADOR = 17418  
DC\_SPANISH\_HONDURAS = 18442  
DC\_SPANISH\_NICARAGUA = 19466  
DC\_SPANISH\_PUERTO\_RICO = 20490  
DC\_SWEDISH = 1053  
DC\_SWEDISH\_FINLAND = 2077  
DC\_THAI = 1054  
DC\_TURKISH = 1055  
DC\_UKRAINIAN = 1058  
DC\_VIETNAMESE = 1066