




4D データベースを 4D v11 SQL に変換する

内容

4D データベースを 4D v11 SQL に変換する	1
必要な動作環境	8
以前のデータベースを 4D v11 SQL に変換	8
変換ウィザード.....	9
ストラクチャファイル情報:.....	10
データファイル情報:.....	10
セグメント情報:.....	11
ディスク情報:.....	11
オプション:.....	12
変換時に問題がある場合.....	12
変換時の動作:.....	13
新しいファイル.....	13
新しいフォルダ.....	13
変換後に Maintenance & Security Center を使用する.....	13
ストラクチャとデータファイル間のリンク.....	15
変換後に消去されるもの:.....	15
変換後に現れる可能性のあるもの :.....	15
Orphan method.....	15
サポートされないもの	16
ユーザモード.....	16
4D リソースとシステムリソースへの直接アクセス.....	16
代替のシステム:.....	17
4DK# - STR# - cicc - キーボードショートカット - マクロ - ピクチャ.....	17
サブテーブル.....	17
旧来のコンポーネント.....	18
データセグメント.....	19
4D 2003/2004 マクロ.....	20
新しいコマンド:.....	21
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.....	21
テーブルのアクセス権.....	21
背景ピクチャのモード選択.....	21
.pth ファイルから 4Dlink への置き換え.....	21
“Extras”フォルダは“Resources”フォルダに.....	23
サポートされない機能	23
動作の変更	24
テーブルフォームとプロジェクトフォーム.....	24
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.....	24
テーブルとフィールドの削除が可能に.....	24
トランザクション.....	24

ネストしたトランザクション	25
インデックス	25
新しい"キーワードを含む"演算子	26
セット	27
命名セレクション	27
作成された場所による命名セレクションとセットの可視性:	27
クリップボードとペーストボード	28
メニュー	28
編集メニューの特別な機構	28
階層リスト	29
リストボックス	29
トリガ	29
プラグイン	30
Plugin SDK	30
バーチャルストラクチャ	31
ピクチャ	31
ピクチャ変数の型宣言	31
ピクチャフィールドへの情報格納:	31
環境設定の互換性:	32
デザイン互換性:	32
Web 互換性:	33
プラットフォーム	33
新しい互換性オプション:	34
外部からのドラッグアンドドロップを拒否	34
トランザクションのネストを許可する	34
Unicode モード	34
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	36
数値フォーマットにシステム設定を使用する:	38
QUERY BY FORMULA をサーバ上で実行する	38
ORDER BY FORMULA をサーバ上で実行する	38
QUERY BY FORMULA で SQL Join を使用する	38
サポートされない互換性オプション:	38
コマンド	39
新しいコマンド	39
Unicode サポート関連:	39
SQL エンジンと SQL サーバ関連:	39
ネストしたトランザクション関連:	39
クエリと並び替え関連:	39
プロジェクトフォーム関連:	40
ドラッグ&ドロップ関連:	40
ペーストボード関連:	40

メニュー関連:	41
階層リスト関連:	41
リストボックス関連:	41
プリント関連:	41
4D 環境関連:	41
システム環境関連:	42
ユーザインターフェース関連:	42
メソッド関連:	42
文字列、XLIFF、正規表現関連:	42
ストラクチャ定義関連:	42
Web 関連:	42
XML 関連:	42
ピクチャ関連:	42
ツール関連:	43
シンタックスが変更されたコマンド:	43
メニュー関連:	43
階層リスト関連:	43
リストボックス関連:	43
プリント関連:	43
コンポーネント関連:	43
4D 環境関連:	44
システム環境関連:	45
データ入力とユーザインターフェース関連:	45
メソッド関連:	46
文字列関連:	46
ストラクチャ定義関連:	47
Web 関連:	47
XML 関連:	47
システムドキュメント関連:	47
BLOB 関連:	48
リソース関連:	48
通信関連:	48
Web サービス関連:	48
プロセス関連:	48
名称が変更されたコマンド:	48
ペーストボード関連:	49
メニュー関連:	49
階層リスト関連:	49
ストラクチャ定義関連:	49
配列関連:	49
動作が変更されたコマンド:	49

第一引数にテーブルが必須となるコマンド:	52
テーマが変更されたコマンド:	52
最適化されたコマンド:	53
廃止されたコマンド:	53
実行しても意味のないコマンド:	53
無効なシンタックス:	54
サブレコードコマンド:	54
統合	54
4D Insider のオブジェクト移動と検索・置換:	54
4D Insider の検索は“デザインモードを検索”に:	56
結果ウィンドウ	57
検索やり直しボタン	57
見つかったオカレンスの数	57
名称変更	57
4D Tools は MSC に	57
4D Pack:	58
廃止された 4D Pack コマンド:	58
コマンドは存在するが使用を推奨しないもの:	59
向上したコマンド:	59
変更のないコマンド:	60
新しいコマンド	60
ショートカット	60
デザインモード	60
レコード	61
Appendix 1: ランゲージコード:	63
Appendix 2: 4DPop	66
インストール	66
利用する	66
利用可能なコンポーネント:	68
4DPop Bookmarks	68
4D Pop Color Chart	68
4D Pop Commands 	68
4DPop Constants Editor	68
4DPop Image Buddy	68
4DPop Migration	68
4DPop Rulers	69
4D sqlSchemas	69
4DPop Window	69
4DPop XLIFF	69
4DPop Quiz	70

4DPop Bug Report.....	70
Chromo 4D	70
B3dCreate.....	70
RegexLab	71
Appendix 3: 4D v11.1 SQLの新機能	72
4D View の Unicode サポート	72
メニューとタイトルにおける Unicode サポート(メニューエディタとプログラミング).....	72
SQL: 大文字小文字を区別した比較 と区別しない比較 (環境設定参照).....	72
ポインタ型の式:.....	72
クイックレポートエディタで 4D の印刷設定で動作.....	72
ピクチャのクリック座標:.....	72
3つの新しいコマンド:.....	73
シンタックスの変更:.....	73
Appendix 4: 4D v11.2 SQLの新機能	74
あたらしい Welcome ウィザード	74
フィールドインスペクタ内の新しいフィールドプロパティ.....	74
新しいメソッド属性: サーバ上で実行.....	74
リストボックス:.....	74
リソースエクスプローラ:.....	74
Web エリア:.....	74
4D Server:.....	74
プログラミングに関する新機能	75
フォルダの新しい場所:.....	75
SET DATABASE PARAMETER の新しいセレクタ.....	76
新しい引数を受け入れるコマンド:.....	76
Web	76
新しい振る舞い:.....	76
4D Internet Commands.....	77
4D View.....	77
Appendix 5: 4D v11.3 SQLの新機能	78
マルチコア最適化	78
Windows Server 2008 に対応.....	78
4D アプリケーション間の Web サービス最適化.....	78
4D for OCI が MacIntel 互換に.....	78
新しい SVG コンポーネント	78
Buildapp XML keys の非サポート	78
SQL.....	78
“WEDD”リソースの非サポート.....	78
4D 2004 データベースから 4D v11 への手順.....	78
プログラムによるエラー管理.....	78
Web サービスを使用したデータベース間のデータ交換.....	79
XML コマンドの再構成.....	79

ドキュメントのサイズが最大 2 GB に	79
リストボックス	79
OK 変数の更新	79

必要な動作環境

Mac: iMac 以上 / Mac OS X 10.4.5 および 10.5

512 MB RAM 以上 - 1 GB 推奨

4D v11.0 SQL: Certified version: 10 4.10

4D v11.1 SQL: Certified version: 10 4.10 および 10.5

4D v11.2 SQL: Certified version: 10 4.11, 10.5.2 および 10.5.3.

4D v11.3 SQL: Certified version: 10.4.11 および 10.5.5

Windows: Pentium II 以上 / Windows XP

512 MB RAM 以上 - 1 GB 推奨

4D v11.0 SQL: Certified version: Windows Server 2003 R2, Windows XP SP2, Windows VISTA

4D v11.1 SQL: Certified version: Windows XP SP2, Windows VISTA

4D v11.2 SQL: Certified version: Windows XP SP2/SP3, Windows VISTA SP1, WinServ2003 R2 SP2

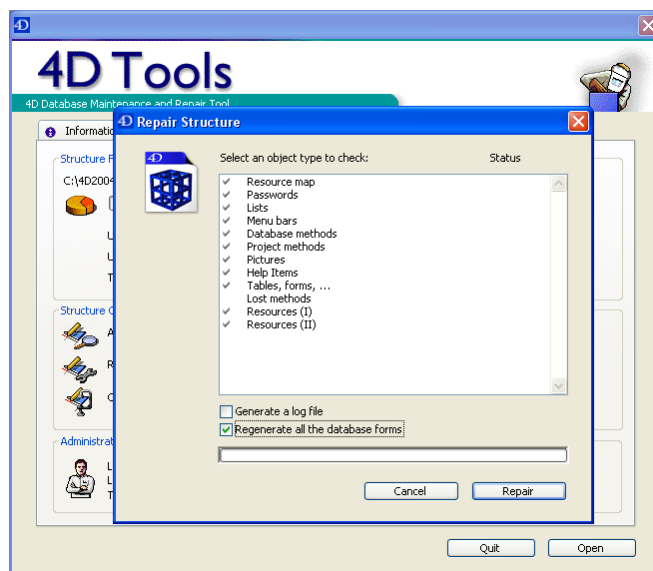
4D v11.3 SQL: Certified version: Windows Server 2008

(ただし Windows Server 2008 の新しいインストールオプションである“Server Core”には非対応)

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:



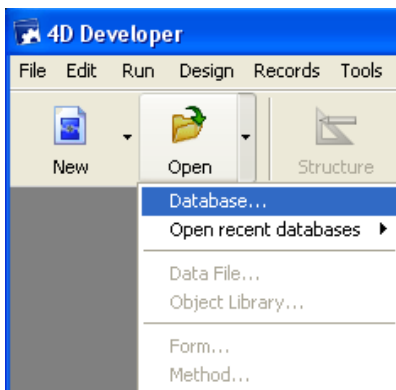
データベースがそれよりも古いバージョンのものである場合、“失われたメソッド”オプションを有効にして、ストラクチャをチェックします。

メソッド解析時の問題を避けるため、Ctrl を押しながら 4D 2004 でストラクチャを開くか、ファイルを開くダイアログで"ストラクチャファイルをチェック"オプションを選択し、"データベースを再トークナイズ"オプションにチェックを入れます。

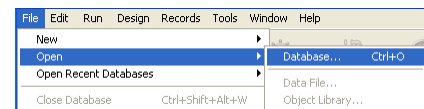
4D v11 SQL を起動します:



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:

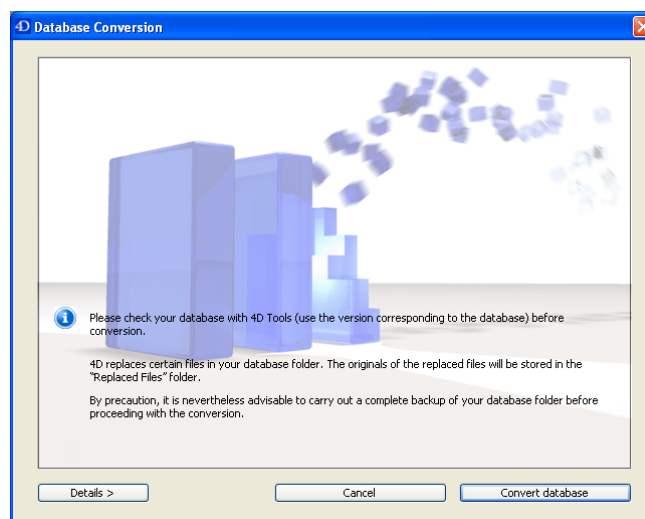


または開くメニューを使用します:



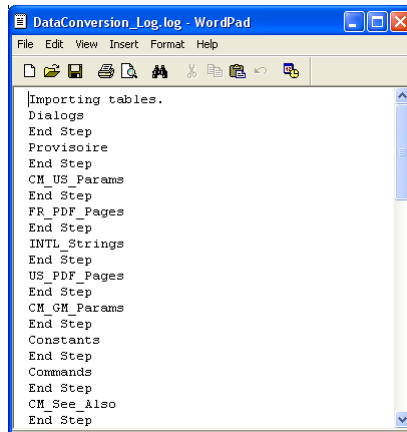
変換する.4DB ファイルを選択します。すると変換ウィザードが表示されます。

変換ウィザード



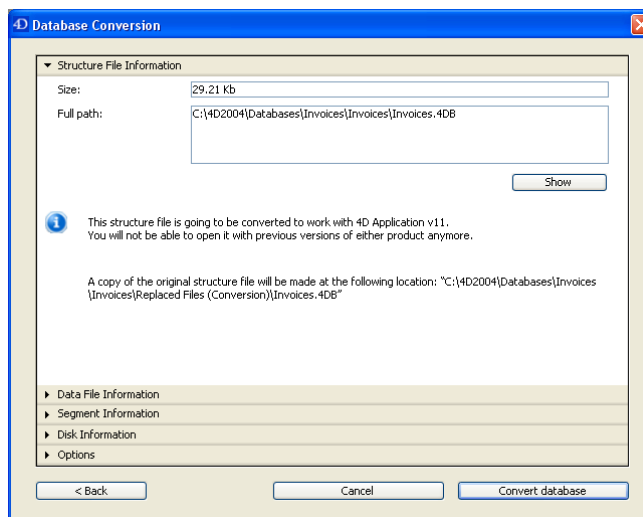
変換ウィザードが提供されます。ウィザードは元のデータベース (ストラクチャ、データ、ログファイル、および空のログファイル) のコピーを、"Replaced files (Conversion)" というフォルダに作成します。

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。

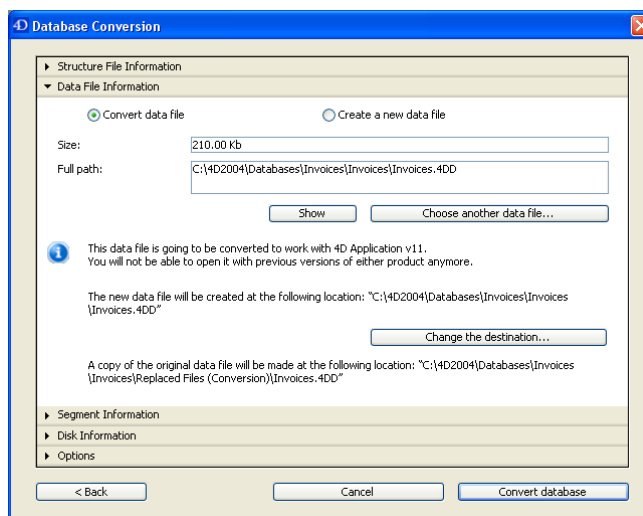


このファイルには、以下のエラーが記録される可能性があります:
 1012: 1 レベルを超えるサブテーブルを持つデータベースを変換しました。

ストラクチャファイル情報:

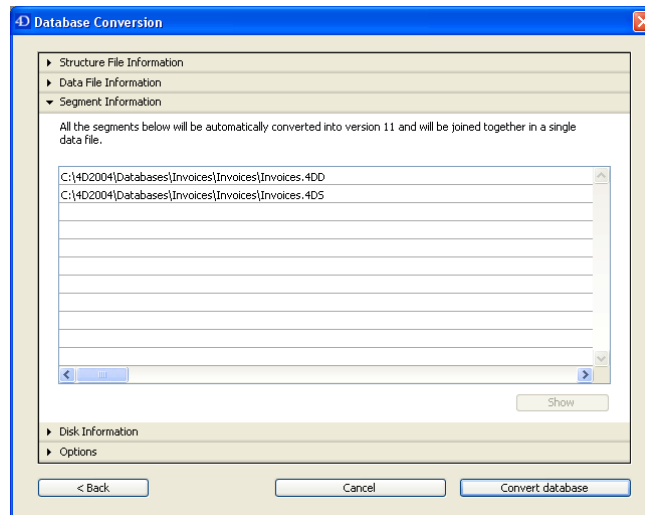


データファイル情報:



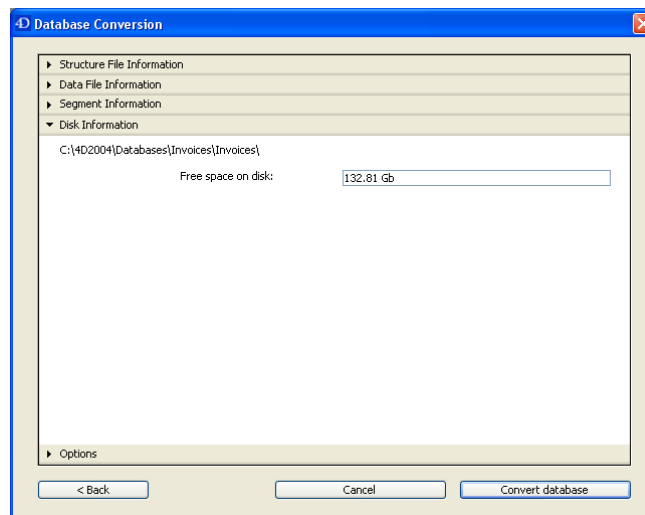
新しいデータファイルを作成できます。
 複数のデータファイルを変換できます。変換ダイアログはその都度表示されます。

セグメント情報:



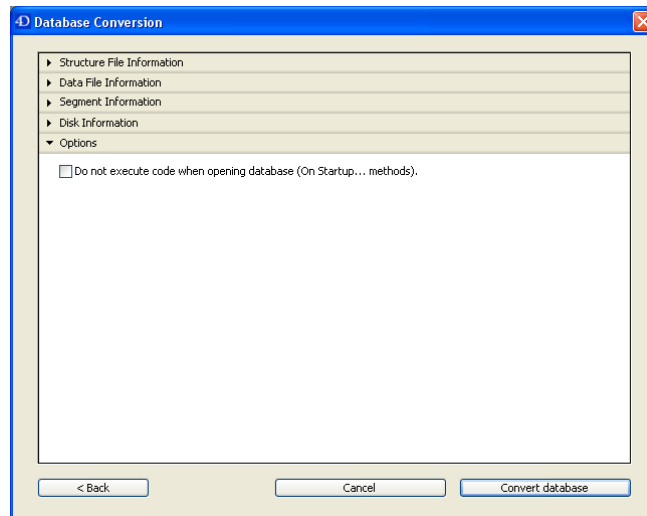
4D v11 SQL にはデータセグメントが存在しません。変換を行うと、すべてのセグメントはひとつにグループ化されます。すべてのセグメントがそろっていないければ、変換を行うことはできません。

ディスク情報:



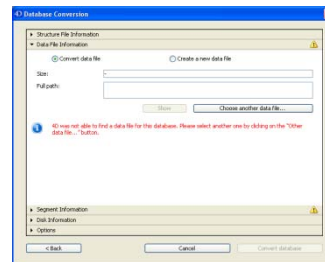
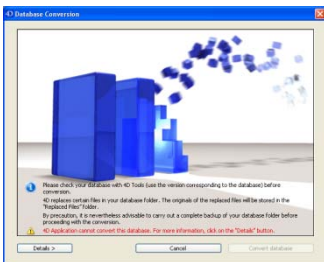
オプション:

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)。
- インター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 のインタープリタモードは 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。

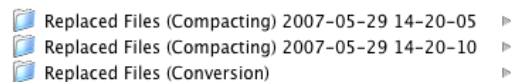
注: 4D v11 SQL で作成されたデータベースは、4dbase フォルダに格納されます。

Mac OS では、データベースはパッケージとして表示されます。変換したデータベースをパッケージにするには、フォルダに、4dbase 拡張子を付け、必要なファイルをすべてこのフォルダに置きます。

新しいフォルダ

- Replaced files:

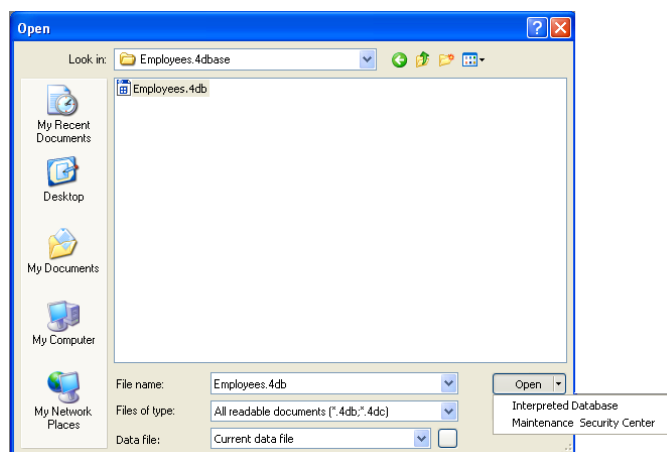
いくつかの操作により、**Replaced files** という名前のフォルダが作成されます: これらのフォルダには、変換前や圧縮前、または修復前のもとのファイルのコピーが納められます。



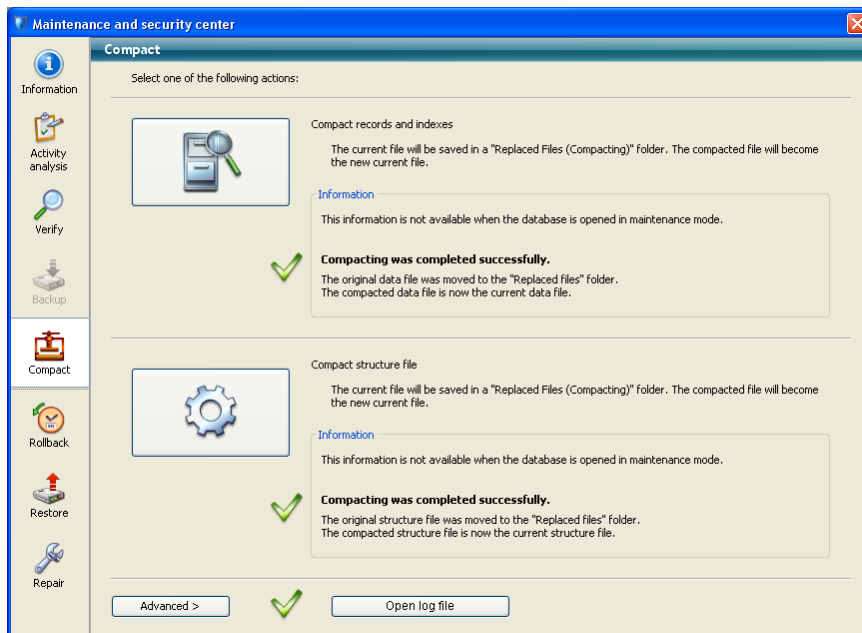
これらのファイルをバックアップとしては使用できないことに留意してください。変換に関連するファイル (.4DA, .4DB, .4DD, .4DS など) しかコピーされません。

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

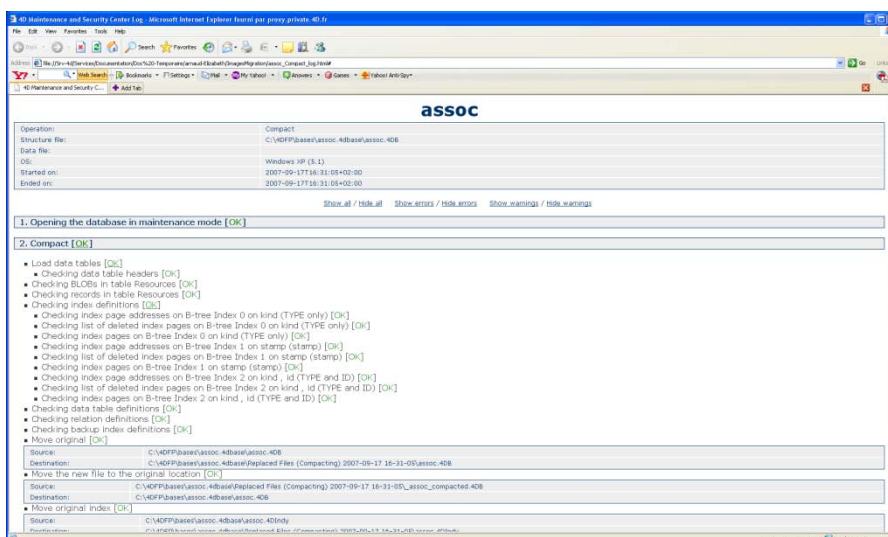
変換後に Maintenance & Security Center を使用する



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 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:



ストラクチャとデータファイル間のリンク

4D v11 SQL でデータベースを変換あるいは作成するたびに、それぞれのデータベースにストラクチャとデータのリンクを設定する ID が割り当てられます。同じ 4D 2004 のデータベースを 2 回 4D v11 に変換すると、同じ ID は割り当てられません。2 番目の.4DD ファイルを 1 番目の.4DB で開くことはできません。データファイルとストラクチャが対応していない旨の警告が表示されます。

(テーブルの削除など) 非互換を引き起こすようなストラクチャに変更がなければ、変換済みのストラクチャファイルと未変換のデータファイルを v11 で開くことができます。データファイルの変換のために変換ダイアログが表示されます。

また変換済みのデータファイルを未変換のストラクチャファイルとリンクさせることもできます。変換ダイアログでデータファイルを選択し、その後変換がおこなわれます。

11.3 では WEDD リソースは使用されません。

警告: 4D v11 SQL ではテーブルやフィールドを削除できるようになりました。ストラクチャとデータを別個に変換し更新する必要がある場合は、一貫性を保つよう留意してください。テーブルやフィールドの追加に問題はありますが、削除は避けてください。削除はデータとストラクチャ両方の変換が終了してから行います。

複数回の変換を行う場合、ストラクチャファイルと同階層に作成される"catalog.xml"が使用されます。

変換後に消去されるもの:

- デバッグモードの設定
- "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
- フローチャートモードのメソッド

変換後に現れる可能性のあるもの :

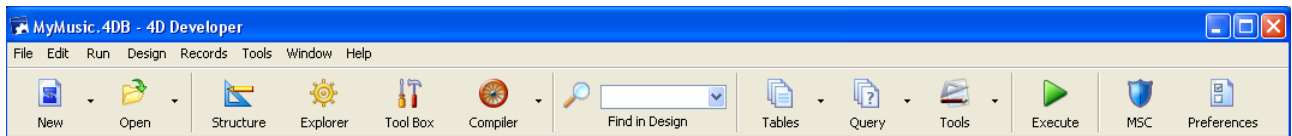
Orphan method

メソッドリスト中に orphan method (迷子のメソッド) が表示されることがあります。これらのメソッドは削除してかまいません。実際これらは以前に削除されたオブジェクトのオブジェクトメソッドです。

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コンセプトを使用しないよう、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
```

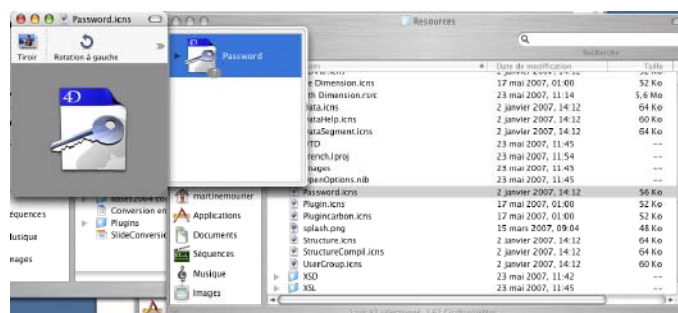
```
$Resources :=Open resource file ($Structure)
```

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、いくら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。この場合、**Open resource file** を使用しなければなりません。

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# - cicc - キーボードショートカット - マクロ - ピクチャ

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。
- ピクチャは.png に変換されます。
- その他

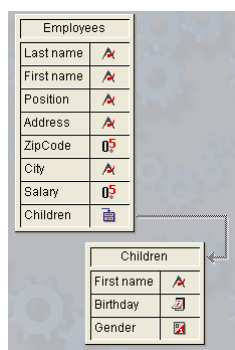
このコンポーネントに関する詳細は appendix3 を参照してください。

サブテーブル

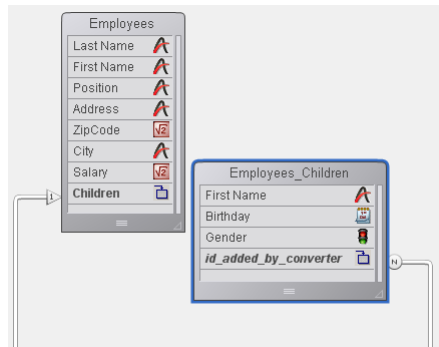
4D v11 SQL では、サブテーブルを作成することができません。

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、特別なリ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:



これは以下のようになります:



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り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りレーションを削除すると、もう元には戻せません。再度りレーションを貼るには標準のりレーションを使い、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、り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。

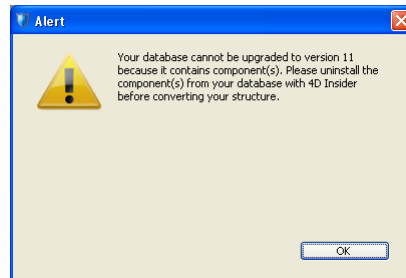
他方、今バージョンからは、**Field** と **Field name** コマンドはポインタ参照が使用された場合、サブフィールドの番号や名前を返します (メインテーブルのそれではなく)。

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ープリタで、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せん。)

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:

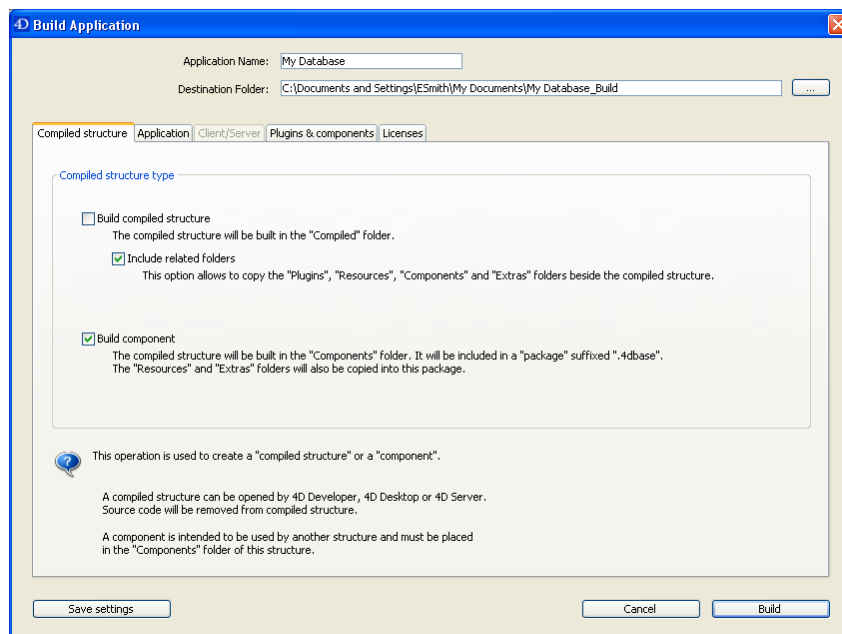


以前のバージョンのコンポーネントは、変換前に、元のバージョンの 4D Insider を使用してアンインストールしなければなりません。

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: .4db、.4dc または .4dbase。プロフェッショナルアプリケーションでは .4dbase 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:

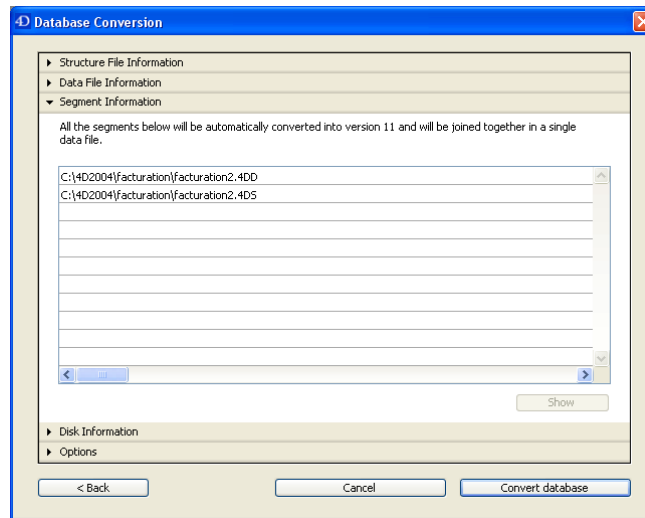


データセグメント

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す。

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 < や > 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

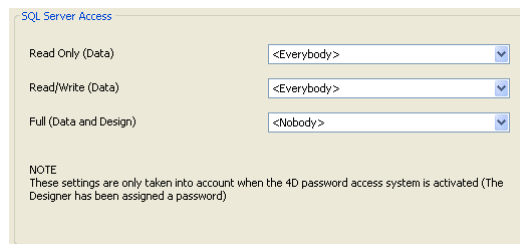
リソースを使用したログインウィンドウのアイコンのカスタマイズ

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。

.pth ファイルから.4Dlink への置き換え

.4DLINK ファイルは XML ファイルで、.pth ファイルに比べ柔軟性がより高いデータベースアクセスファイルです。一つのタグ<database_shortcut>があり、いくつかの属性を設定できます。以下は利用可能な属性のリストです:

すべてのタイプのデータベース:

属性名	説明	利用できる値	デフォルト値
is_remote	データベースがローカルかリモートかを指定する	True False	False
user_name	ユーザ名	テキスト	“”
password	パスワード	テキスト	“”
md5_password	ハッシュ化されたパスワード	テキスト	“”
structure_opening_mode	0: 通常 1: インタプリタ 2: コンパイル済み	0, 1, 2	0

リモートモードで開かれるデータベース:

属性名	説明	利用できる値	デフォルト値
server_database_name	サーバデータベースの名前 (拡張子なし)	テキスト	“”
server_path	IP アドレスまたは DNS 名	テキスト	“”
open_login_dialog	接続ダイアログを表示するよう 4D にリクエストする	True False	False

ローカルモードで開かれるデータベース:

属性名	説明	利用できる値	デフォルト値
open_in_custom_mode	アプリケーションモードでデータベースを開く	True False	False
open_tools	MSC (Maintenance Security Center) でデータベースを開く	True False	False
create_structure_file	4D にストラクチャファイルを作成するよう指示する	True False	False
structure_file	ストラクチャファイルのパス名	テキスト	“”
create_data_file	4D にデータファイルを作成するよう指示する	True False	False
data_file	データファイルのパス名	テキスト	“”
skip_onstartup_method	4D に On Startup データベースメソッドをスキップするよう指示する	True False	False

definition_import_file	XML ストラクチャ定義ファイルのパス名	テキスト	“”
resources_import_file	Internal use only		
data_opening_mode	1: デフォルトデータファイル 2: 他のデータファイルを選択 3: 新しいデータファイルを作成	1, 2, 3	1

“Extras” フォルダは “Resources” フォルダに

カスタムデータの転送はストラクチャファイルと同階層の“Extras”フォルダを使用して行われていました。このフォルダの利用は今後推奨されません。“Resources”フォルダを利用するようにしてください (Resources エクスプローラ参照)。

互換性を肘するため、“Extras”はまだサポートされています。

サポートされない機能

- プラグインをピクチャフィールドに格納すること (Blob フィールドに置き換えられました)。
- 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。
- 名前を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:
Count tables	Get last table number
Count fields	Get last field number

追加されたコマンド:
Is table number valid
Is field number valid

警告: ストラクチャとデータファイルを個別に変換し更新する必要がある場合、一貫性を保持するように留意してください。テーブルやフィールドの追加に問題はありませんが、削除はストラクチャとデータをともに変換した後に行うべきです。

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。

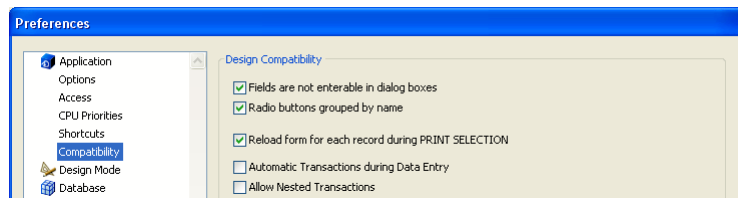
レコード番号に対して使用するコマンド (**SCAN INDEX**、**RELATE ONE SELECTION** 等) をトランザクション中で利用可能になります。

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。

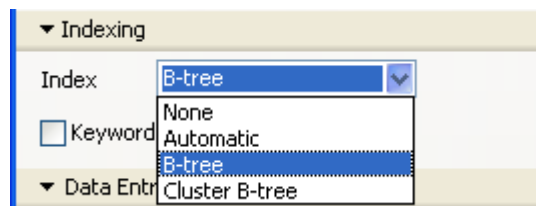


このオプションは 4D v11 SQL の SQL エンジンによるトランザクションには影響しません。SQL エンジンのトランザクションは常にマルチトランザクションです。

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。

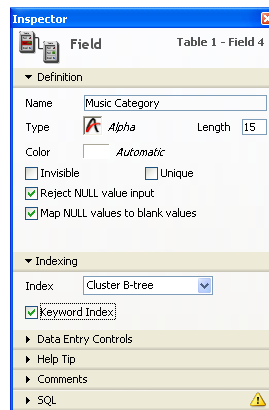
- **標準の B-tree:** 多目的のインデックス。以前のバージョンの 4D で使用されていました。
- **クラスタ B-tree:** データ中に同じ値が繰り返し表れるときに有効なインデックス (例: ブールフィールドや、Mr. Mrs. Ms. で構成される敬称など)。
- **自動 (デフォルト):** データに基づき 4D がどのインデックスを使用するか決定します。

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

Btree Index#10 on TECHNOTES.RefAuteur
 Checking Index Pages Addresses on Btree Index#10 on TECHNOTES.RefAuteur (OK)
 Checking List of deleted Index Pages on Btree Index#10 on TECHNOTES.RefAuteur (OK)
 Checking Index Pages on Btree Index#10 on TECHNOTES.RefAuteur (2 errors)
 Type# Description
 24 127 :
 24 127 :

Btree Index#11 on TECHNOTES.Title
 Checking Index Pages Addresses on Btree Index#11 on TECHNOTES.Title (OK)
 Checking List of deleted Index Pages on Btree Index#11 on TECHNOTES.Title (OK)
 Checking Index Pages on Btree Index#11 on TECHNOTES.Title (2 errors)
 Type# Description
 24 127 :
 24 127 :

- キーワードインデックス:

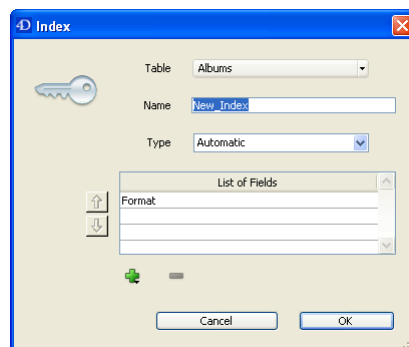


この新しいタイプのインデックスは、文字型またはテキストフィールドで利用できます。標準のインデックスと同時に利用できます。4Dはコンテキストに応じて、より効率の良いインデックスを使用します。

現時点で、キーワードインデックスはICUのライブラリを使用してテキストを解析しインデックス化します。日本語のような単語の切れ目がいまいちな言語では、正しいインデックスを作成することはできません。

- 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。

命名セレクション

名前が\$から始まるローカル命名セレクションを作成できるようになりました。ローカルセットと同様、ローカル命名セレクションはそれを作成したプロセス内でのみ使用でき、サーバ上では使用できません (下表参照)。

作成された場所による命名セレクションとセットの可視性:

	クライアントプロセスで作成			サーバプロセスで作成		
	\$test	test	<>test	\$test	test	<>test
クライアントプロセス	可視	可視	可視			
他のクライアントプロセス			可視			
他のクライアント						
サーバプロセス		トリガ		可視	可視	可視
他のサーバプロセス						可視

クライアントプロセスで作成: 可視



サーバプロセスで作成: 可視

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)
MENU BAR	SET MENU BAR
HIDE MENU BAR	"ユーザインターフェース"テーマに移動
SHOW MENU BAR	"ユーザインターフェース"テーマに移動
SET ABOUT	"ユーザインターフェース"テーマに移動
SET MENU ITEM KEY	SET MENU ITEM SHORTCUT

変更なし:

Menu selected

Count menus

Count menu items

DISABLE MENU ITEM

ENABLE MENU ITEM

APPEND MENU ITEM

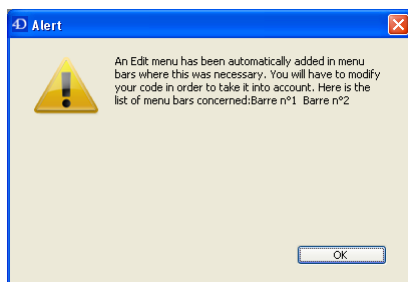
INSERT MENU ITEM

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。

リストボックス

特定のフォームイベントにおいて、**Focus object** コマンドから返される値は、コマンドがリストボックスあるいはリストボックスの列のオブジェクトメソッドから呼ばれた場合、4D 2004 と 4D v11 で異なります:

以下はその違いです:

フォームイベント	Focus object コマンドから返される値	
	4D 2004	4D v11 SQL
On Clicked	列へのポインタ	リストボックスへのポインタ
On Double Clicked	列へのポインタ	リストボックスへのポインタ
On Column Moved	列へのポインタ	リストボックスへのポインタ
On Column Resize	列へのポインタ	リストボックスへのポインタ
On Header Click	ヘッダへのポインタ	リストボックスへのポインタ
On After Sort	ヘッダへのポインタ	リストボックスへのポインタ

利用可能なデータソース: 配列の他にカレントセレクションや命名セレクション

隠し行を指定する配列の処理

複数行のセル (Windows のみ).

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。

大きな動作の変更:

	4D 2004	4D v11 SQL
サーバ上のトリガ	クライアントと同じカレントレコードとカレントセレクション	当該テーブルのカレントレコードのみがサーバ上のトリガでも保証される

4D v11 SQL より、サーバ上のランゲージコンテキストはクライアントマシンのそれと完全に独立するようになりました。その結果トリガ内で可能な動作にはほぼ制限がなくなりました。他のテーブルに対するクエリ、リレーションの有効化なども可能です。

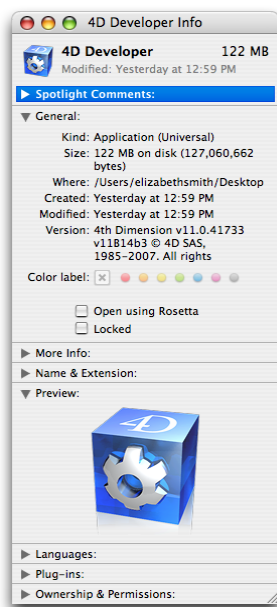
[On Load Record](#) トリガは廃止されました。

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 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 は Rosetta で使用しなければなりません。(4D for OCI は 4D v11 SQL Release 3 より MacIntel ネイティブです。) MacIntel 上で非ネイティブプラグインを使用すると、"XX.bundle をプラグインとしてロードできません"というエラーが表示されます。この場合プラグインが Intel アーキテクチャをサポートしていないことが考えられます。

MacIntel 上で非ネイティブプラグインを使用するには、Finder 上で 4D.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

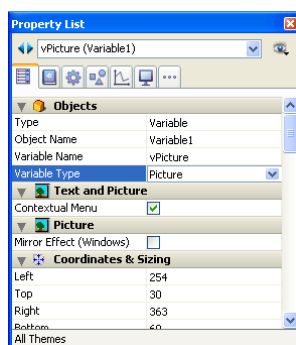
ピクチャはネイティブフォーマットで保存されます。

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:



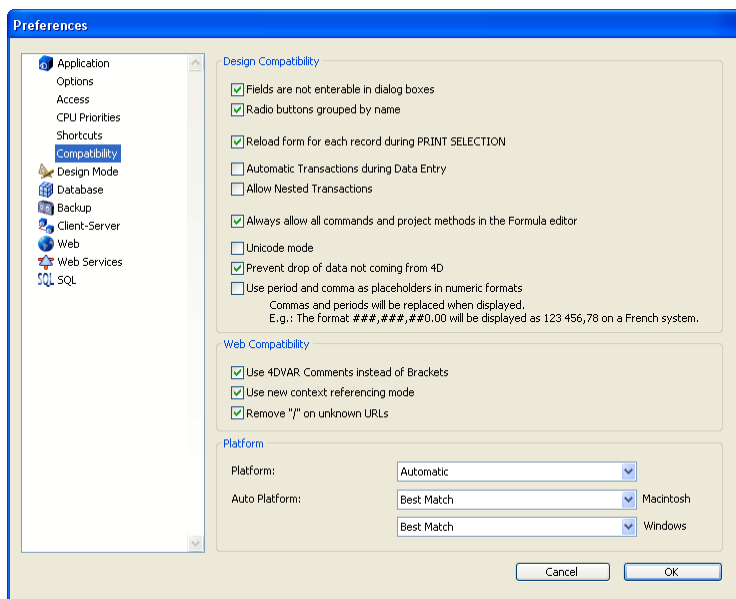
- またはコンパイラ宣言 **C_PICTURE(vPict)** を使用する。この場合、フォームがロードされる前にコードが実行されていなければなりません。これは **On Load** フォームイベントより前でということです。

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でグループ化する:
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が使用されています。
- **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勧めしません。**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勧めします。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

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。

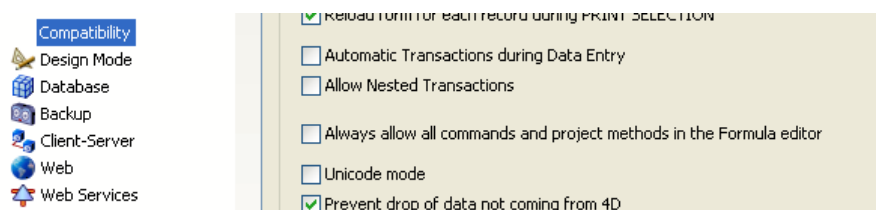
環境設定:

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
C_TEXT	テキスト型	テキスト型
C_STRING	文字変数型	テキスト型 文字長引数は無視
ARRAY TEXT	テキスト配列型	テキスト配列型
ARRAY STRING	文字配列型	テキスト配列型 文字長引数は無視
Char	Char (Ascii コード) 0-255	Char (UTF-16 値) 0-65535
Ascii は Character code に変更	Character code ASCII コードを返す	Character code Unicode コードを返す
Length, String, Position, Delete string, Substring, Change string 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。
Mac to Win, Win to Mac, Mac to ISO, ISO to Mac		動作しません

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

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

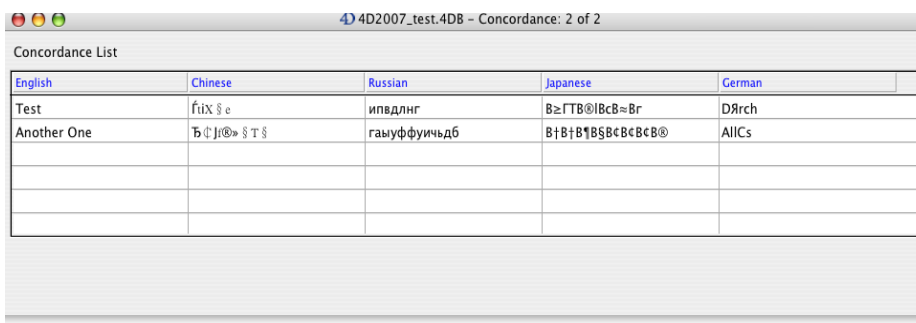
3;4;1032

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	ипедлнг	В≥ГТВ@lBcB≈Br	DЯrch
Another One	Ъ¢J!®»§T§	гауффуичьдб	†††††B§B€B€B€B®	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	ипедлнг	さけおいでゆ	Durch
Another One	哈搭亞三卅	гауффуичьдб	ああえういいお	Allés

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。

QUERY BY FORMULA をサーバ上で実行する

このオプションがチェックされていると、**QUERY BY FORMULA** と **QUERY SELECTION BY FORMULA** コマンドはサーバ上で実行されます。

ORDER BY FORMULA をサーバ上で実行する

このオプションがチェックされていると、**ORDER BY FORMULA** コマンドはサーバ上で実行されます。

QUERY BY FORMULA で SQL Join を使用する

このオプションがチェックされていると、**QUERY BY FORMULA** と **QUERY SELECTION BY FORMULA** コマンドは SQL のように JOIN を実行します。この場合ストラクチャ上のリレーションは必要ありません。

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

2004 クリップボード

4D v11 SQL ペーストボード

"com.4d.text.native"	ネイティブ文字のテキストファイル
"com.4d.text.utf16"	Unicode文字のテキストファイル
"com.4d.text.rtf"	リッチテキストファイル
"com.4d.picture.pict"	PICT画像
"com.4d.picture.png"	PNG画像
"com.4d.picture.gif"	GIF画像
"com.4d.picture.jfif"	JPEG画像
"com.4d.picture.emf"	EMF画像
"com.4d.picture.bitmap"	BITMAP画像
"com.4d.picture.tiff"	TIFF画像
"com.4d.picture.pdf"	PDF画像
"com.4d.file.url"	アクセスファイル

これらの 4D シグネチャはクロスプラットフォームです。

あるいは UTI (Uniform Type Identifier、Mac OS のみ) やフォーマット番号や名前 (Windows のみ) も使用できます。 .

警告: 互換性のため 4 文字のタイプ (TEXT, PICT, etc.) も保持されています。Mac OS では“TYPE”と“CREATOR”に正しい情報が書かれているかに注意してください。

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ィンドウを表示
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 44: SQL Engine Case Sensitivity 45: Client Log Recording 46: Query By Formula On Server 47: Order By Formula On Server 48: Auto Synchro Resources Folder 49: QUERY BY FORMULA Joins 50: HTTP Compression Level 51: HTTP Compression Threshold	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ト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";*) -> Ê :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";*) -> ÊÀ :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
```

Replace string (source; oldString; newString{; howMany} {; *}) -> String

Replace string コマンドは最後の引数にアスタリスクを受け入れます。この引数を渡すと、大文字小文字やアクセント文字を区別するようになります。言い換えれば文字コードベースでの比較を行うようになります。(a # A, a # à, など).

実際、変換されて Unicode モードにしたデータベースでは、以前のバージョンと同じ動作をさせるためにはアスタリスクを渡さなければなりません。この場合、文字列比較は文字コードを使用して行われます。これはアクセント文字だけでなく改行やタブ、Char(0)などを考慮する場合も同様です。

以下の例題で * 引数の役割を示します:

```
vResult:=Replace string("Crème brûlée","Brulee";"caramel") `vResult は "Crème caramel"
```

```
vResult:=Replace string ("Crème brûlée";"Brulee";"caramel";*) `vResult は "Crème brûlée"
```

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 と **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:

Web server Process	Longint	-13
Execute on Client Process	Longint	-14
4D Server Process	Longint	-15
On Quit Process	Longint	-16
Method editor macro Process	Longint	-17
Internal 4D Server Process	Longint	-18
Backup Process	Longint	-19
Log File Process	Longint	-20
Restore Process	Longint	-21
MSC Process	Longint	-22
Timer Process	Longint	-23
SQL Method Execution Process	Longint	-24
Server Controller Process	Longint	-25
Monitor Process	Longint	-26

名称が変更されたコマンド:

EXECUTE

Find index key

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)。

Ascii
GET PLUGIN LIST
USE ASCII MAP

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 はポイントされていないフィールドを受け入れなくなりました。

Version type: 4D 2004 では standard バージョンで 0 を、demo バージョンで 1 を返していました。4D v11 SQL では、この関数は常に 0 を返します。これは 4D の試用モードが変更されたためです。今バージョンではプラグインを含むすべての機能を 1 カ月だけ利用可能になりました。

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。

警告: クライアント/サーバでは、Web フォルダの場所が変更になりました (4D 2004 ではデフォルトで実行可能ファイルと同階層): 4D v11 SQL では実行可能ファイルと同階層ではなく、ストラクチャのローカルコピーと同階層のフォルダ内です (異なるクライアントからの複数のドメインを提供する場合のため)。

他方、4D 2004 のようにサーバの Web フォルダの配信は自動では行われません。フォルダを正しい位置にコピーするか **SET HTML ROOT** コマンドを使用しなければなりません。

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 はインター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。

GOTO RECORD: 4D 2004 と異なり、このコマンドはレコード番号が削除されたレコードに対応する場合、エラーを生成するようになりました。このエラーは **ON ERR CALL** でとらえることができます。

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

REGISTER CLIENT / EXECUTE ON CLIENT: これらのコマンドは最適化され、ネットワーク通信が格段に減少し、多数の 4D クライアントマシンが接続した状態でも使用ができるようになりました。

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:

- プラグインデータを格納するためにはピクチャではなく Blob を使用してください。
- **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ーシ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。

このなりレーシ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:

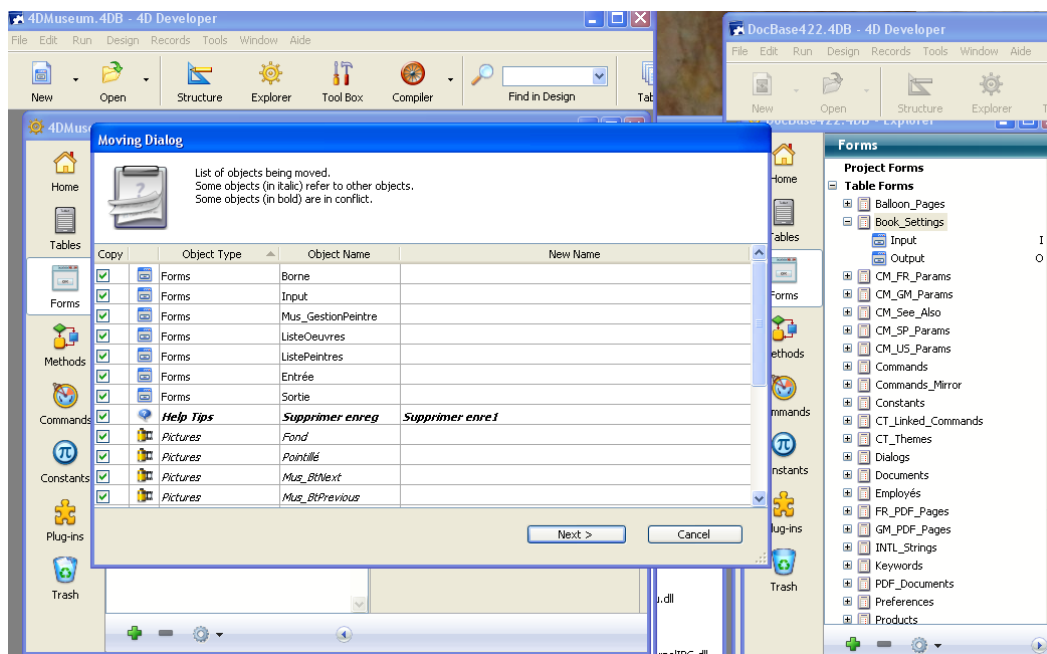
エクスプローラのホームページからテーブルやフォーム、メソッドなどをドラッグ&ドロップできます
以下のケースでは移動ダイアログが表示されます:

- 名前の衝突
- “常に表示”オプションがデータベース環境設定で選択されているとき
- “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場合

データベース間でのオブジェクト移動 (4D Insider の機能の一部を統合)

- 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)



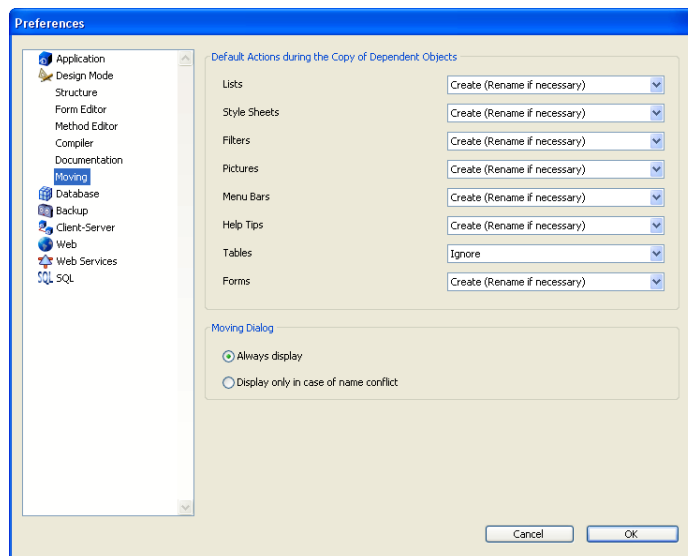
名前の衝突がある場合は赤で表示されます。

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。

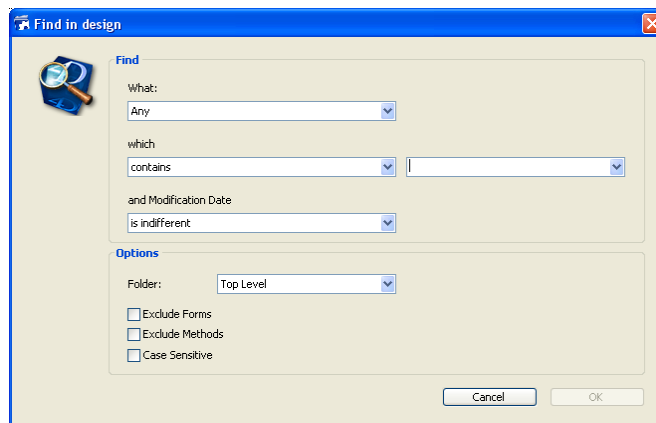
すべてを選択または選択解除するには、ALT+クリック (Win) OPTION + クリック (Mac) します。他のオブジェクトを使用: 依存オブジェクトにのみ存在します。

環境設定で移動オプションを設定できます:



4D Insider の検索は “デザインモードを検索” に:

編集メニュー > デザインモードを検索:



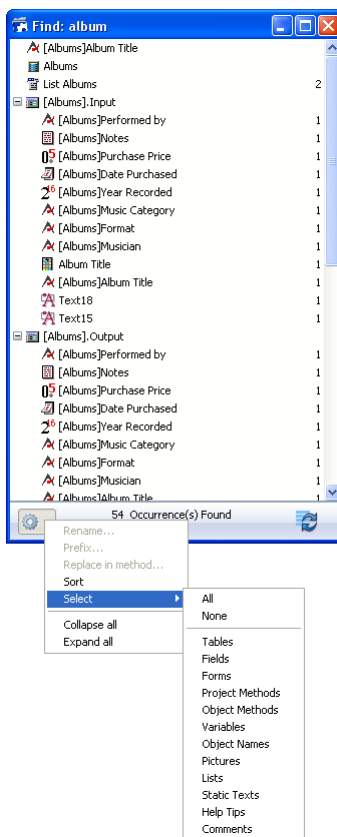
タイプ: 式, 変数, テーブル/フィールド, フォームオブジェクト, コメント, プロジェクトメソッド (この場合 “呼び出し元” と “呼び出し先” リスト)

検索条件: 等しい, 含む, 前方一致, 後方一致 (@利用可能)

更新日

フォルダ

検索のスコープ: フォーム, メソッド, メニュー, 選択リスト, テーブル/フィールド, コメント, ピクチャライブラリ, テキスト, ヘルプ tips, フォーマット/フィルタ, スタイルシート, プラグインコマンド, 4D コマンド, 定数 および フォルダ.



結果ウィンドウ

オプション: 名称変更, 接頭辞, メソッド内を置換 (1 つ以上), 並び替え, 選択 (Shift+Click または CTRL/CMD+Click), すべてを折りたたむ/すべてを展開する

検索やり直しボタン

見つかったオカレンスの数

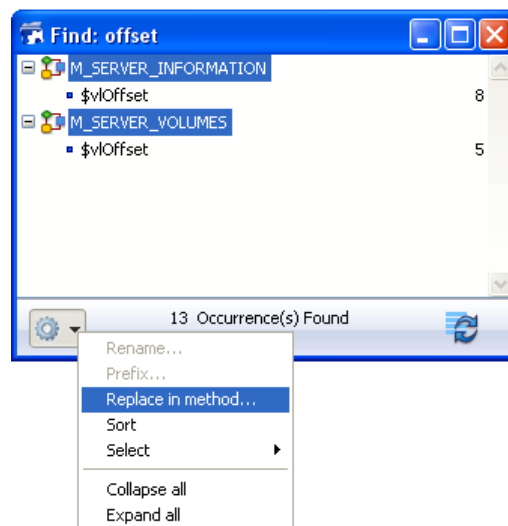
“呼び出し元”と“呼び出し先”リストはプロジェクトメソッドにのみ表示されます。

名称変更

変数, プロジェクトメソッドおよびフォーム

オプション: ソースオブジェクトも置き換える (“呼び出し元”)

メソッド中も置き換える: ひとつあるいは複数のメソッドを検索して置き換えるために使用できます。

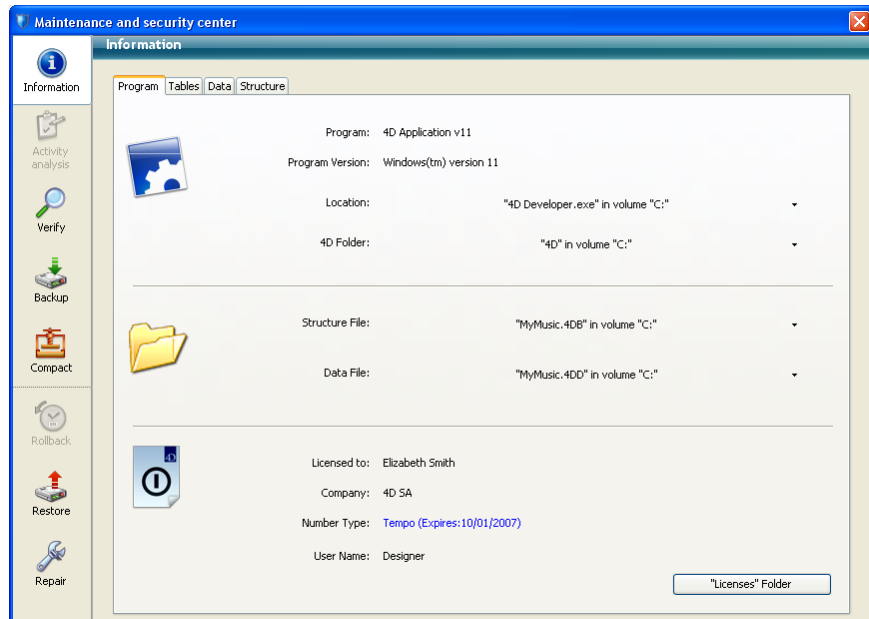


新しいコンポーネント管理と、XLIFF 利用を容易にする 4DPop XLIFF も参照してください。

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勧めしません。v11 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:

AP PICT DRAGGER

AP PICT UPDATER

AP Read Picture File

AP Save BMP 8 bits, AP Save GIF

AP Read Picture BLOB

AP PICT DRAGGER, AP Rect dragger

AP SET PICT MODE

AP Select document

AP Set palette

AP ShellExecute

AP Sublaunch

4D 2004 ですでに廃止されたコマンド:

AP PrintDefault

以下に置き換え:

4D のドラッグ&ドロップ

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

READ PICTURE FILE

CONVERT PICTURE および **WRITE PICTURE FILE**

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

MouseDown, MouseX および MouseY システム変数

SET FORMAT を使用

Select document を使用

Select RGB color を使用

LAUNCH EXTERNAL PROCESS を使用

LAUNCH EXTERNAL PROCESS を使用

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
AP Add table and fields

SET DATABASE PARAMETER を使用
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るよう検討中です)

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 PARAMSET 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)
```

AP Does method exist

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

Appendix 2: 4DPop



Note: 以下のリストは完全で保障されたものではありません。今後追加されたり取り除かれたりすることがあります。

4Dpop はデザインモードにパレットをインストールします。必要に応じて互換性のあるコンポーネントをインストールし、このパレットにツールを追加できます。

4Dpop は無料でダウンロードできます

<http://www.4d.com/jp/downloads/products.html>

インストール

1. "Components"フォルダをデータベースのストラクチャと同階層に作成する
2. "4DPop.4dbase"またはそのエイリアスを"Components"フォルダに配置する
3. データベースを開く
4. "On Startup"データベースメソッドに"Install 4DPop"とタイプし、タブキーを押す。メソッドに以下のコードが挿入されます。

```
`Run 4Dpop  
`This code is available even if the component is not present like  
`in the final application.
```

If (Not(Is compiled mode))

```
ARRAY TEXT($arrComponents_Txt;0)  
COMPONENT LIST($arrComponents_Txt)
```

```
If (Find in array($arrComponents_Txt;"4DPop")>0)  
EXECUTE METHOD("4DPop_Palette")
```

```
End if
```

```
End if
```

5. データベースを再起動します。デザインモードに入ると画面の左下にパレットが表示されるようになります。

* Macintosh ではコンポーネントはパッケージであり、システム設定によっては".4dbase"拡張子が表示されないことがあります。

利用する

4DPop のヘルプは"ヘルプ"メニューからアクセスできます。

パレットが閉じられている場合、ハンドルを引っ張ることで開くことができます。



パレットが閉じられているときにハンドルをダブルクリックすると、現在インストールされているツールをすべて表示できる大きさにパレットは拡張します。パレットが開かれているときにダブルクリックすると閉じられます。

互換性のあるコンポーネントがインストールされていないと、パレットは以下のピクチャを表示します:



一つ以上の互換コンポーネントがインストールされていると、それらはボタンとして表示されます。

ボタンをクリックすると、ツールが起動されるかメニューが表示されます。

Option+click (Macintosh) または Alt+click (PC) するとコンポーネントに割り当てられたヘルプファイルが開かれます。(ヘルプファイルはコンポーネントストラクチャと同階層に“MyComponent.htm”ファイルとして配置されます。)



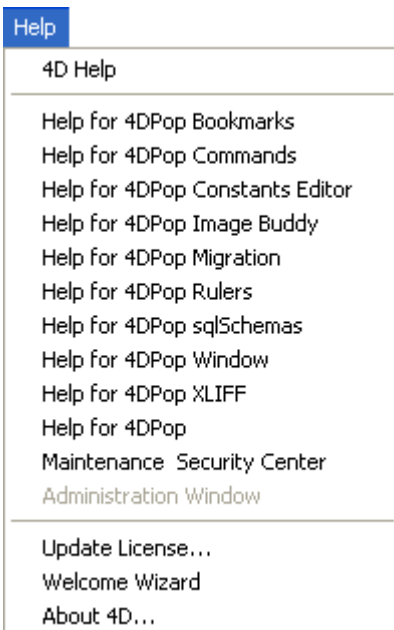
タイトルバーをクリックして、右や左にパレットを移動できます。その位置は記憶されます。

パレットのタイトルバーにコンテキストメニューが割り当て可能です。そこにはすべてのツールがグループ化され、利用可能なパレットのオプションが表示されます。

コンポーネントはアルファベット順にロードされます。特定のコンポーネントを先にロードし、左に表示させるには、フォルダ名に例えば 01 を追加します。

ヘルプ

4D のヘルプメニューからコンポーネントのヘルプにアクセスできます。



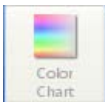
利用可能なコンポーネント:

4DPop Bookmarks



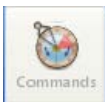
指定したページ、電子メール、ファイル、フォルダ、サーバなどにデザインモードから直接素早くアクセスできます。

4D Pop Color Chart



機能的な(RGB, HSL, etc.)パレットを使用してオブジェクトで使用する背景色と描画色を選択し、対応するソースコードをドラッグアンドドロップでメソッド中に生成します。

4D Pop Commands



4D コマンドのドキュメントへのアクセスを提供します。またメソッドエディタにドラッグアンドドロップすることでパラメタリスト付きでメソッドに挿入できます。

4DPop Constants Editor



4DPop Constant Editor を使用してカスタム定数を作成できます。

4DPop Image Buddy

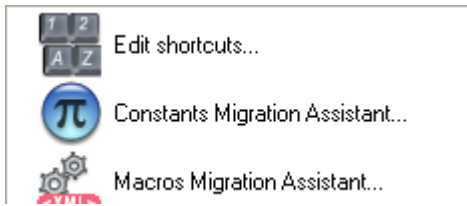


4DPop Image Buddy は“Resources”フォルダにある画像を管理するためのツールです。

4DPop Migration



4D v11 SQL への移行を手助けするツールです:



デザインモードのショートカットを編集する
リソースフォークのユーザ定数を専用のプラグイン形式に移行する
マクロファイルを変換するなどが行えます。

4D Pop Rulers



アプリケーションやデザインモードでルーラやガイドを表示します。

4D sqlSchemas



グラフィカルに SQL スキーマを設定するために使用します。

4D Pop Window



デザインおよびアプリケーションモードのウィンドウを一括して管理するためのツール。ウィンドウの並び替えやオフスクリーンウィンドウの表示などが可能です。

4D Pop XLIFF



データベースの XLIFF ファイルの管理や編集を行い、またテキスト参照をメソッドエディタやフォームエディタにドラッグできます。

互換コンポーネントを作成する方法は 4D Pop コンポーネントヘルプを参照してください。

オープンソース

このコンポーネントはコンパイルバージョンで提供されますが、コンポーネントの“Sources”フォルダ内にソースコードを見つけることができます。

4DPop Quiz



キーワードでコマンドを探す、ダブルクリックでオンラインドキュメントにアクセスする、メソッドエディタにコマンドをドラッグする。

4DPop Bug Report



デザインモードの中からバグレポートが可能です。基本的な情報は入力済みで、説明欄にソースコードをドラッグアンドドロップで挿入できます。

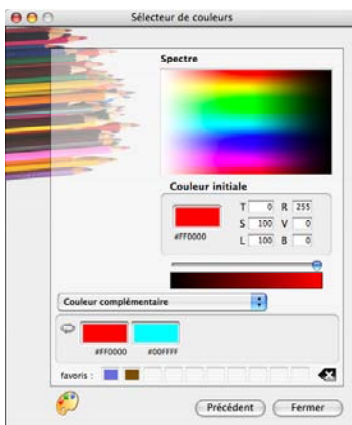
Chromo 4D



Chromo4D は 4D v11 を使用する開発者のために作成されました。

このコンポーネントを使用して開発者は、スペクトラムあるいは開発者が選択したピクチャから色を選択できます。

提供される組み合わせは色に関する昔からの理論に基づいて選択されます。スペクトラムをクリックして選択されたテーマの色の組み合わせを見ることができます。



B3dCreate

<http://forums.4d.fr> から取得できます。

一つのピクチャから 4 状態 (アクティブ、クリック、ロールオーバー、無効) のボタンピクチャを作成するために使用します。

RegexLab



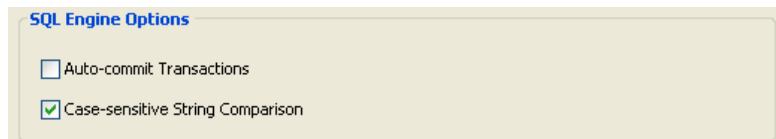
RegexLab は 4 つの機能を提供します。Regex_Match、Regex_Substitution、Regex_Split、Regex_Extract。
またラボを使用して Regex において様々な検索パターンを試すことができます。

Appendix 3: 4D v11.1 SQL の新機能

4D View の Unicode サポート

メニューとタイトルにおける Unicode サポート(メニューエディタとプログラミング)。

SQL: 大文字小文字を区別した比較 と 区別しない比較 (環境設定参照)



を設定するデータベースパラメタ セレクタ = 44 (SQL Engine Case Sensitivity)

ポインタ型の式:

SQL クエリで直接ポインタ型の式を参照可能 (逆参照されているかされていないかにかかわらず)。

逆参照されたポインタ変数を渡す:

```
C_LONGINT ($vLong)
C_POINTER ($vPointer)
$vLong:=1
$vPointer:=-->$vLong
Begin SQL
SELECT Col1 FROM TEST WHERE Col1=:$vPointer->;
End SQL
```

逆参照されていないポインタ変数を渡す:

```
C_LONGINT ($vLong)
C_POINTER ($vPointer)
$vLong:=1
$vPointer:=-->$vLong
Begin SQL
SELECT Col1 FROM TEST WHERE Col1=:$vPointer;
End SQL
```

この原則はすべてのタイプの SQL クエリで使用できます: ODBC コマンド、**Begin SQL/End SQL** タグ、**QUERY BY SQL** コマンド

Note: 1 レベルのポインタのみが使用できます。ポインタのポインタは使用できません。

クイックレポートエディタで 4D の印刷設定で動作

ピクチャのクリック座標:

4D でピクチャフィールドや変数上でのクリック座標を MouseX と MouseY 取り出すことができるようになりました。これはスクロールやズームも考慮されます。座標はピクセル単位で、ピクチャの左上が(0,0)です。このメカニズムが正しく動作するためには、表示フォーマットが“トランケート、中央合わせなし”でなければなりません。このメカニズムはマップピクチャと同様のもので、スクロール可能なボタンバーや地図ソフトウェアのインタフェースに使用できます。

3つの新しいコマンド:

DESCRIBE QUERY EXECUTION (status)

Get Last Query Plan (descFormat) -> 文字列

Get Last Query Path (descFormat) -> 文字列

DESCRIBE QUERY EXECUTION (True) `解析モード開始

ARRAY TEXT (aTitles;0)

ARRAY TEXT (aDirectors;0)

Begin SQL

```
SELECT CONCAT(CONCAT(ACTORS.FirstName,' '),ACTORS.LastName),
    CITIES.City_Name
FROM ACTORS, CITIES
WHERE ACTORS.Birth_City_ID=CITIES.City_ID
ORDER BY 2,1
INTO :aTitles, :aDirectors;
```

End SQL

vResultPlan:= **Get Last Query Plan** (Text format description)

vResultPath:= **Get Last Query Path** (Text format description)

DESCRIBE QUERY EXECUTION (False) `解析モード終了

このコード実行後、**vResultPlan** と **vResultPath** には実行されたクエリの説明が格納されます:

vResultPlan:

```
[Join] : ACTORS.Birth_City_ID = CITIES.City_ID
```

vResultPath:

And

```
[Merge] : ACTORS with CITIES
```

```
[Join] : ACTORS.Birth_City_ID = CITIES.City_ID (227 records found in 0 ms)
```

```
> 227 records found in 0 ms
```

シンタックスの変更:

Replace string (source; oldString; newString{; howMany} {; *}) -> 文字列

Replace string コマンドは最後の引数にアスタリスクを受け入れるようになりました。この引数を渡すと、文字の評価は大文字小文字や発声記号を区別するようになります。いいかえれば大文字小文字やアクセント文字を区別します:

(a#A, a#â など)。

変換され Unicode モードに変更されたデータベースでは、以前と同じ動作をさせるためにはアスタリスクを渡さなければなりません。この場合、文字比較は文字コードベースで行われます。これはアクセント文字だけでなく、改行やタブ、Char(1)や Char(0)を扱う場合も同様です。

*引数の役割を例示します。:

```
vResult:=Replace string("Crème brûlée";"Brulee";"caramel") `vResult is "Crème caramel"
```

```
vResult:=Replace string ("Crème brûlée";"Brulee";"caramel";*) `vResult is "Crème brûlée"
```

Appendix 4: 4D v11.2 SQL の新機能

あたらしい Welcome ウィザード

フィールドインスペクタ内の新しいフィールドプロパティ

自動インクリメント: 正数、倍長整数、Integer 64 bits

内部ストレージの最大サイズ: テキスト

新しいメソッド属性: サーバ上で実行

サーバ上に新規プロセスは作成されない (**Execute on server** コマンドと異なる点)

4D Server では実行をリクエストしたクライアントプロセスの対応プロセスが使用される。

パラメタと戻り値の転送はメソッドが通常通り呼ばれたのと同様に、自動で行われる。

変数へのポインタの場合 (変数、配列、配列要素)、参照される値もサーバに送信されます。参照された値がサーバ上で実行されたメソッドで変更されると、更新された値がクライアントに送信され、クライアント側の値も更新されます。テーブルやフィールドのポインタは参照として送信されます(テーブル番号、フィールド番号)。

カレントレコードの値は自動では好感されません。

Note: このオプションはインタプリタモードとコンパイルモードで同様に動作します。

リストボックス:

配列表示モードの際の**隠し行配列**プロパティ。

複数行セル (Windows のみ)

リソースエクスプローラ:

“Resources”フォルダの中身を閲覧するために使用されます。追加、削除要素の閲覧などを行うことができます。またクライアント/サーバモードでリソースの自動更新も行えます。

Web エリア:

フォーム上に直接 Web コンテンツを表示するために使用するオブジェクトです。HTML ページ、ファイル、ピクチャ、Javascript、Flash、PDF、さらに Windows 上では MS Office ドキュメントも表示できます (MS Office がインストールされていること)。

Note: 4D Web 2.0 Pack の 4D Live Window を置き換えます。

4D Server:

様々な最適化が図られています。以下のドキュメントを参照してください:

<http://www.4d.com/jp/docs/SVC/SVCMDIP.HTM>

プログラミングに関する新機能

新しいデータベースメソッド: [On SQL Authentication](#)

OPEN ADMINISTRATION WINDOW

NOTIFY RESOURCES FOLDER MODIFICATION

QUERY SELECTION WITH ARRAY

TRUNCATE TABLE

WA OPEN URL

WA Get current URL

WA OPEN BACK URL

WA Back URL available

WA OPEN FORWARD URL

WA Forward URL available

WA REFRESH CURRENT URL

WA STOP LOADING URL

WA Execute JavaScript

WA EXECUTE JAVASCRIPT FUNCTION

WA SET URL FILTERS

WA GET URL FILTERS

WA SET EXTERNAL LINKS FILTERS

WA GET EXTERNAL LINKS FILTERS

WA GET LAST URL ERROR

WA Get last filtered URL

WA GET URL HISTORY

WA Create URL history menu

WA Get page title

WA SET PAGE CONTENT

WA Get page content

WA SET PAGE TEXT LARGER

WA SET PAGE TEXT SMALLER

WA SET PREFERENCE

WA GET PREFERENCE

フォルダの新しい場所:

active 4D folder の新しい場所

Windows Vista: C:\Users\{ user_name }\AppData\Roaming\4D

Windows XP: C:\Documents and settings\{ user_name }\Application Data\4D

Mac OS X: /users/{user_name}/library/Preferences/4D

Licenses Folder の新しい場所

Windows Vista: C:\ProgramData\4D\Licenses

Windows XP: C:\Documents and settings\all users\Application Data\4D

Mac OS X: /library/application support/4D/licenses

SET DATABASE PARAMETER の新しいセレクタ

Selector = 45 (Client Log Recording)

Selector = 46 (Query By Formula On Server)

Selector = 47 (Order By Formula On Server)

Selector = 48 (Auto Synchro Resources Folder)

Selector = 49 (QUERY BY FORMULA Joins)

新しい引数を受け入れるコマンド:

LOG EVENT は Mac OS で動作し、新しい引数を受け入れます。

BLOB TO PICTURE は第 3 の引数 *codec* を受け入れます。

READ PICTURE FILE は最後の引数にアスタリスクを受け入れすべてのファイルタイプを読み込めます。

GET LISTBOX ARRAYS は隠し行配列を受け取るための新しい引数を受け入れます。

Web

OPEN WEB URL

Unicode の完全サポート

URL またはファイルパス名を含めることができる

Mac OS では追加のプロトコル (file:, mailto:, news:, http:, etc.) をサポート

Mac OS ではコロン':、Windows ではバックスラッシュ'\、または file:// から始まる POSIX URL を受け入れます。この場合、4D はシステムに最も適したアプリケーションでファイルを開くよう要求します (例えば.html ファイルの場合ブラウザ、.doc ファイルの場合 MS Word など)。この場合*引数は無視されます。

OPEN WEB URL ("http://www.4d.com")

OPEN WEB URL ("file://C:/Users/Laurent/Documents/pending.htm")

OPEN WEB URL ("C:\Users\Laurent\Documents\pending.htm")

OPEN WEB URL ("mailto:jean_martin@4d.fr")

AUTHENTICATE WEB SERVICE と **SET WEB SERVICE OPTION**: DIGEST モードによる Web サービスの認証。

SOAP DECLARATION: Web サービスとして XML 構造を送信 (WSDL の変更が必要)

新しい振る舞い:

QUERY BY FORMULA(aTable{; queryFormula})

QUERY SELECTION BY FORMULA (aTable {; queryFormula })

これらのコマンドは SQL と同様の JOIN を実行します。これによりストラクチャ定義上での自動リレーションは存在しなくても検索が可能です。

On record loading トリガイベントは今後サポートされません。

4D Internet Commands

SMTP_SetPrefs

15 : UTF-8 & quoted-printable

16 : UTF-8 & base64

4D View

新しいバージョンの 4D View ではエリアのペインを固定できます。

PV FREEZE PANES

PV UNFREEZE PANES

Appendix 5: 4D v11.3 SQL の新機能

マルチコア最適化

Windows Server 2008 に対応

4D アプリケーション間の Web サービス最適化

4D for OCI が MacIntel 互換に

新しい SVG コンポーネント

ドキュメント: <http://www.4d.com/jp/docs/SGC/SGCMDUS.HTM>

Buildapp XML keys の非サポート

Buildapp XML keys "/Preferences4D/BuildApp/OEM/Interface/ServerAboutPictMac" と "/Preferences4D/BuildApp/OEM/Interface/ServerAboutPictWin"は v11.3 よりサポートされなくなりました。4D 開発者は **SET ABOUT** コマンドを使用してアプリケーションについてダイアログをカスタマイズできません。

SQL

スキーマのサポート: テーブルを含むオブジェクト (つまりデータベース自身!)

4D データベース間の直接接続

ODBC コマンドの名称変更を伴う、コマンドの再構成 ("外部データソース"テーマは"SQL"テーマに統合)

Flash Player リクエストのサポート (環境設定>SQL - "socketpolicy.xml"ファイルの追加)。

SQL コードで複数行の挿入

"WEDD" リソースの非サポート

データベースの変換あるいは作成時に、4D が内部的なリンク番号を発行。

4D 2004 データベースから 4D v11 への手順

4D 2004 以前のデータベースを 4D v11 SQL に変換する際に、"catalog.xml"ファイルが生成されるようになります。このファイルは将来、同じデータファイルを使用できるようストラクチャを再変換する際に使用できます。

プログラムによるエラー管理

ON ERR CALL のフレームワークにおいて、**GET LAST SQL ERROR** は **GET LAST ERROR STACK** に名称が変更されました。今後はこのコマンドで Web サービス、XML、XSLT エラーの処理も行えます。ODBC や SQL エラーについては **SQL GET LAST ERROR** を使用しなければなりません。

Web サービスを使用したデータベース間のデータ交換

SET WEB SERVICE OPTION ([Web Service HTTP Compression](#); [Web Service Deflate Compression](#))

新しいセレクタ: 50 と 51

SET DATABASE PARAMETER (HTTP compression level;-1) 1 (速度優先) ~ 9 (圧縮率優先). -1 = 最適設定

XML コマンドの再構成

XML DOM: DOM コマンドのグループ

XML SAX: SAX コマンドのグループ

XML ユーティリティ: XSLT、エラー、SVG など汎用コマンドのグループ

DOM EXPORT TO PICTURE は **SVG EXPORT TO PICTURE** に名称変更

2つの新しいコマンド:

DOM Get Root XML element

SVG element ID by coordinates: 指定された座標にある要素の ID 属性値を返す(“id”または“xml:id” 属性)。

ドキュメントのサイズが最大 2 GB に

扱うことのできるドキュメントの最大サイズが 2GB になりました。

Get document size, **SET DOCUMENT SIZE** そして **SET DOCUMENT POSITION** コマンドが関連します。

リストボックス

Mac OS で複数行の入力と表示が可能に (Windows では既に可能でした)。

OK 変数の更新

統計関数 (**Average**, **Sum**, **Sum squares**, **Variance**, **Std deviation**, **Max**, **Min**) を大量のデータに対して使用する時に (ゆえに進捗サーモメータが表示される場合)、ユーザが停止ボタンをクリックすると **OK** 変数に 0 が設定されるようになりました。