

アップグレードガイド

4D 2004



はじめに	4
アップグレードガイド 4D 2004	5
1 - データベースの変換	5
一般的な注意事項	5
作業を開始する前に	6
用意するものと変換に必要なツール	6
エクスターナルおよびプラグインの扱い	6
バージョン 3.5 から 2003 までのプラグイン	7
非常に初期のバージョンを変換する(バージョン 2 および 3)	7
2 - アップグレードの手順 (Copy-Compact-Convert)	9
アップグレードによって露呈したデータベースの不具合に対処する	9
3 - 4D 2004	12
ストラクチャとデータ: リソースフォークとデータフォークの分離	12
新しいアーキテクチャとオペレーションシステム	13
プラグインフォルダの互換性	13
新しいフォルダ	13
設定フォルダの新しい場所	13
新しいプラグインアーキテクチャ	14
新しいインストール場所	14
プラグイン用の新しいフォルダ	14
クライアント/サーバでプラグインをダウンロードする	15
読み込みの優先順位	15
ライセンスのアクティベーション	15
4 - 変換されたデータベース: 互換性	16
ストラクチャの互換性	16
Web 互換性	17
メニューバー	17
プラットフォーム (環境設定およびフォームプロパティ)	17
フォームウィンドウのサイズ	18
システムハイライトカラー	18
ASCII テーブル編集	18
プラグインの対応	19
5 - 統合されたアプリケーション	19
統合された 4D Insider の機能	19
統合された 4D Customizer Plus の機能	20
プラグインに統合された 4D Customizer の機能	21
追加されたパラメータ	22
4D Backup の統合と新しいバックアップメカニズム	22
統合された 4D ODBC	23
4D 2004 で追加されたプラグイン	24
ユーザ&グループの管理	24
6 - 動作面で新しくなったこと	24
フォーミュラエディタ	24
ユーザモード、カスタムモード、サブフォームで表示されたリストフォームの動作を統一: すべてのリストフォームに適用される選択モード	24
リストフォーム関連	25
リストフォーム関連の新イベント	25
リスト関連の新コマンド	25
リストフォーム関連	25
リスト更新関連	25
サブフォーム関連	26
7 - 新しいキーボードショートカット	27

変更されたキーボードショートカット	27
追加されたキーボードショートカット(デザインモード)	28
カスタムモードからユーザモードに切り替える方法	28
8 – 4D 2004 以降で新しくなったこと	28
ランゲージで新しくなったこと	29
追加されたコマンド	29
変更されたコマンド	29
名称が変更されたコマンド	29
追加された定数	29
追加されたフォームイベント	30
バックアップで新しくなったこと	30
リストボックスで新しくなったこと	30
4D Tools で新しくなったこと	30
4D Write で新しくなったこと	31
4D View で新しくなったこと	31
4D Pack で新しくなったこと	31
コンポーネント	32
9 – リストフォーム Q&A	32
基本コンセプトの覚え書き	32
4D 2004 のサブフォームプロパティおよび過去バージョンとの互換性	33
単一選択リストでカレントレコードがロードされないのはなぜですか。	33
なぜウインドウの再描画でリストフォームのカレントレコードがアンロードされるのですか。	33
表示されたリストフォームに最初からハイライトされたレコードがあるのはなぜですか。	34
リストフォームに空行が表示されなくなったのはなぜですか。	34
空行ダブルクリックによるレコード追加を避けるべきなのはなぜですか。	34
空行のダブルクリック以外にレコードを追加する方法がありますか。	35
4D 2004 に変換されたリストフォームの選択モードはどうなりますか。	36
On Clicked イベントで自動的にカレントレコードがロードされなくなったのはなぜですか。	36
On Displayed Detail が空行で発生するのはなぜですか。	37
サブフォームを選択しても点滅アイコンが表示されないのはなぜですか。	37
リストフォームオブジェクト(フィールド/変数)が表示されないのですが。	37
リストフォームアピアランスが過去のバージョンと違うのはなぜですか。	38
リスト更新が動作しないように思えるのはなぜですか。	38
フィールドや変数のリスト更新をどうすればコマンドで開始できますか。	38
クリックまたはコマンドでリスト更新モードに突入した行の変数をどうすれば初期化できますか。	39
リストフォームのハイライトをどうすればコマンドで制御できますか。	39
ユーザが単一/複数のレコードをハイライトしたことをコマンドで調べるにはどうすればよいですか。	40
リストフォームで利用できる新しいフォームイベントは何ですか。	40
ファイルプロシージャが実行されないときがあるのはなぜですか。	41

はじめに

このドキュメントは、4D S.A.が過去に発行したテクニカルノート、テクニカルティップス、リファレンスマニュアルなどの内容を再構成したものです。

普通は、複製(Copy)、圧縮(Compact)、変換(Convert)という簡単な手順でアップグレードができます。

このドキュメントでは、データベースを変換するときに注意すべき事柄、変換後に直面するかもしれない問題、互換性の問題、変更が加えられた部分、新しいメカニズムといった内容が網羅されています。

4D 2004では、修正の必要性をできるだけ少なくするため、過去のバージョンとの互換性を保つよう最大限の努力が払われました。アップグレードは、比較的簡単に終わるはずですが。

複雑なケースについては、4D社のテクニカルサポートサービスを利用するという手段もあることを忘れないでください。

アップグレードガイド 4D 2004

1 - データベースの変換

一般的な注意事項

下記の注意事項が守られていれば、データベースを変換することができます:

- バージョン 6 以降は、問題なくアップグレードすることができます。
バージョン 6.5.x データベースをバージョン 2004 に変換する場合、バージョン 6.7、6.8、2003 を経る必要はありません。そのようなデータベースは、直接バージョン 2004 に変換することができます。何らかの不具合が生じるようであれば、中間のバージョンを使用してください。一見、不具合と思われる現象の中には、互換性のオプションを利用するだけで解決できるものもあります。
- バージョン 3 以前は、6.5 あるいは 6.7 のような中間バージョンに一旦アップグレードしてから、バージョン 2003 ないし 2004 にアップグレードしなくてはなりません。
- アップグレードと同時にプラットフォームを変更してはなりません。はじめに元のプラットフォームでアップグレードを完了し、それからプラットフォームを変更してください。
たとえば、Macintosh 版のバージョン 3 を Windows 版のバージョン 2004 に変換する場合、最初にデータベースを Mac 版の 2004 にアップグレードしてから Windows マシンに移行してください。
- 過去のバージョンの 4D を非対応の新しい OS で起動してはなりません。

バージョン 2 または 3 のような過去のバージョンは、明らかに最近の OS で動作するようには作られていません。たとえば、Mac OS 9 以降でバージョン 2 を起動すれば、データレベルで深刻な問題を招く恐れがあります。過去のバージョンを起動するにあたっては、下記の表を参考にしてください:

4D Version	Windows	Mac OS
2.x (*)	N.A.	8.6
3.0.x (**)	N.A.	8.6
3.5.x (**)	Windows 95/98/Millennium/NT3.5/NT4.0	8.6
6.0.5	Windows 95/98/Millennium/NT4.0	8.6
6.0.6	Windows 95/98/Millennium/NT4.0	9.0
6.5	Windows 98/98SE//Millennium/NT4.0	9.0, 9.1
6.7	98SE/Millennium/NT4/2000/XP	9.0 to 9.2.2
6.8	98SE/Millennium/2000/XP	9.1, 9.2.2 to 10.2.2
2003	98/Millennium/2000/XP/2003	9.2.2, 10.2.8, 10.3.9
2003.7	2000/XPsp2/Serv2003	9.2.2(***)/10.2
2004	2000sp4/XPsp1-sp2/Serv2003	10.3.5
2004.2	2000sp4/XPsp1-sp2/Serv2003sp1	10.3.9/10.4.1

(*) 欧州版バージョン 4.x は米国版バージョン 2.x に対応。

(**) 欧州版バージョン 5.x は米国版バージョン 3.x に対応。

(***) CarbonLib 1.4 以降。

この表では、各 OS と 4D バージョンの互換対応レベル(Certified/Compatible)が省かれています。掲載されているのは、最低限の動作環境です。

- データベースをアップグレードするには、各バージョンのシングルユーザ(スタンドアロン)製品を利用します。製品のライセンスは不要です。データベースの変換には、デモ版で充分だからです。
- 過去の 4D および 4D Tools の中間バージョンは、www.4D.com サイトまたはデベロッパネットワークプログラムが発行するライブラリ CD から入手することができます。

作業を開始する前に

- 変換したデータベースは元に戻すことができません。必ずバックアップを作成してください。元のデータベースフォルダをまるごと同じ場所にコピーするようなバックアップはなるべく避けてください。コピーされたストラクチャがオリジナルのデータファイルを開いてしまい、意図に反して変換されてしまうかもしれません。実際、圧縮ユーティリティを使用して元のデータベースを圧縮しておくのが一番安全です。そのようにすれば、確実に使用できるオリジナルが残されることになるからです。
- 変換を開始する前に対応するバージョンの 4D Tools を使用してデータファイルを圧縮するようにしてください。

用意するものと変換に必要なツール

Mac OSのバージョン2または3 データベースをアップグレードする場合、エクスターナルパッケージや外部ルーチンを取り除くためにResEditが必要です。

- ResEdit は、Classic 環境で動作する Mac OS 9 アプリケーションです。

次の場所からダウンロードすることができます:

ftp://ftp.apple.com/developer/Tool_Chest/Developer_Uutilities/ResEdit_2.1.3/

- バージョン2004よりも前のMacintoshデータベースをWindowsプラットフォームに移動する場合、4D Transporterが必要です。
- アップグレードを実行するためには、インタプリタ版のストラクチャが必要であり、パスワードシステムが有効にされているのであれば、Designerのパスワードがなければなりません。

エクスターナルおよびプラグインの扱い

エクスターナル

バージョン3.5以前の4Dでは、プラグインは「エクスターナルパッケージ」として提供されていました。バージョン3.5よりも上位のバージョンに変換されていないデータベースは、アップグレードの障害となるエクスターナルがストラクチャに組み込まれている可能性があります。

そのようなデータベースを上位バージョンに変換する場合、外部ルーチンやエクスターナルが使用する4Dのモジュールはすべて削除しなくてはなりません。

- 4Dのモジュールは、エクスターナルのインストーラを使用して削除することができます。モジュールのインストーラが入手できなければ、ResEditでリソースを削除してください。
- 外部ルーチンは、External Mover または ResEdit で削除することができます。

注記: データベースがエクスターナルをリソースファイル(拡張子.rsr)に含んだ形でWindowsに移植されたのであれば、Mac用に再変換してからエクスターナルを取り除かなくてはなりません。

エクスターナルを取り除く

External Moverを使用する

External Moverを使用する場合、システムの対応状況を確認してください。External Moverは、Mac OS 9以降で実行されるようには作られていないので、そのようなOSで使用するべきではありません。

External Moverは、バージョン2のデータベースを処理することができ、External Mover Plusはバージョン3.5までを処理することができます。ソフトウェアは4Dディスクに収録されています。

External Mover PlusはMac OS 9で起動することができます。External Mover Plusでストラクチャを開くには、optionキーを押しながらストラクチャファイルを選択してください。

ResEditを使用する

重要:最初にデータベースのバックアップを作成し、そのファイルをResEditで開いてください。エクスターナルのあるストラクチャには、特有のリソースタイプが含まれているはずですが、たとえば外部ルーチンやパッケージが使用する次のようなリソースがインストールされているかもしれません: 「4DND」「4DNX」「4DEX」「4DPX」「4DTE」「4Dte」「4DTF」「FON#」「THM#」。上記のリソースは、ストラクチャから削除してください。「PICT」および「STR#」リソースは、アップグレードには影響しないので、削除する必要はありません。

バージョン3.5から2003までのプラグイン

バージョン3.5から2003までのプラグインは、Win4DXおよびMac4DXフォルダにインストールされました。そのようなプラグインを使用するデータベースをアップグレードする場合、プラグインのアップグレードは、ストラクチャやデータファイルのアップグレードとは別個に行なう必要があります。バージョン2004では、プラグインのアーキテクチャが新しくなりました。「新しいプラグインアーキテクチャ」、およびバージョン2004のアップグレードマニュアルを参照してください:

[4D_2004_Upgrade.pdf](#)

このようなプラグインは、ファイルを新しいバージョンに置き換えることによってアップグレードできます。プラグインの開発が終了し、なおかつバージョン2004で動作しないような場合、その機能は失われることになり、関係するすべてのプラグインコールを削除するとともに、可能であれば同等の機能を持つ4Dのコマンドか別のプラグインで代用しなければなりません。無効なプラグインコールを含むストラクチャはコンパイルできません。

プラグインのアップグレード方法は、そのメーカーによって異なります:

- **4D 社製プラグイン(4D Draw, 4D Write など):** 4D から提供される最新版のプラグインをインストールしてください。バージョン2004が存在しないプラグインは、同じ機能が4Dに統合されています。プラグインコマンドを4Dのネイティブコマンドに書き換えてください。
- **サードパーティ社製のプラグイン:** バージョン2004の対応状況をメーカーに問い合わせてください。バージョン6.7以前のプラグインは、基本的にバージョン6.8以降に対応していません。

非常に初期のバージョンを変換する(バージョン2および3)

各バージョンのタイプおよびクリエイター情報

バージョン2および3のデータベースでしばしば問題になるのは、(エクスターナルの件を除けば)バージョン6.5以降で開こうとしても、ダイアログでファイルが薄く表示されてしまい、選択ができないという現象です。これは4Dのバージョンによってファイルのタイプコードおよびクリエイターコードが異なっているためです。次表には、バージョンごとのタイプおよびクリエイター情報が示されています:

4D Version	Structure Type	Structure Creator	Data Type	Data Creator
2.x	BAS3	4D03	data	4D03
3.0	BAS5	4D05	DAT5	4D05
6.5	BAS5	4D06	DAT5	4D06
6.7/6.8	BAS5	4D06	DAT5	4D06
2003	BAS5	4D06	DAT5	4D06
2004	BAS5	4D06	DAT5	4D06

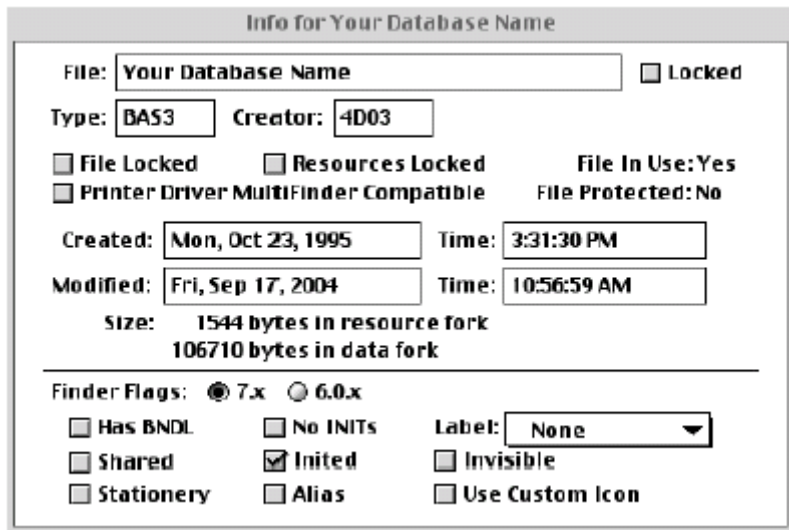
クリエイターコードを変更する

データベースのクリエイターコードは、そのファイルを開いて変換しようとするバージョンのクリエイターコードと一致していなければなりません。クリエイターコードはResEditで変更することができます。

クリエイターコードを変更する方法:

- 1 -ResEdit を起動する。
- 2 -ファイルメニューの「ファイル/フォルダ情報を取得する」を選択する。
- 3 -クリエイターコードを変更したいストラクチャあるいはデータファイルを選択する。

次のようなウィンドウが表示されます:

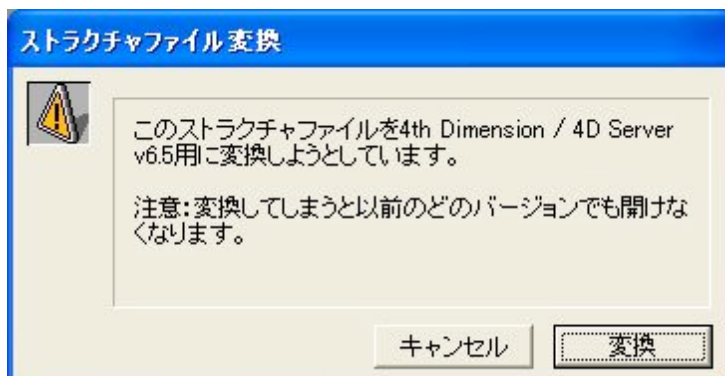


- 4 -仮に、バージョン 6.5 から 6.7 に変換する場合、ストラクチャファイルをタイプ「BAS5」/クリエイター「4D06」に変更し、データファイルをタイプ「DAT5」/クリエイター「4D06」に変更する。
- 5 -変更を保存してから ResEdit を終了する。

タイプおよびクリエイター情報が合っていれば、上位バージョンのデモ版4Dでファイルが開けるようになります。このような方法でバージョン 2 または 3 のデータベースをバージョン 6.7 で開こうとすると、起動時にクラッシュすることがあるかもしれません。そのようなときはひとつ戻ってバージョン 6.5、それでも問題があれば、バージョン 6.0.6r8 にデータベースを変換してください。

バージョン 2 をバージョン 6.5 で開く

- 1 -シングルユーザ版の 4D バージョン 6.5 を起動する。
- 2 -デモ版を選択する。
- 3 -データベースを開く。ストラクチャファイルの変換に関するダイアログを確認する:



4 - 「変換」 ボタンをクリックする。データファイルの変換に関するダイアログを確認する:



データベースがバージョン6.5にアップグレードされます。これが成功すれば、バージョン2004へのアップグレードは問題とされないはずですが。ダイアログメッセージ: 「このデータベースは古い互換モード(BBOX resource)を使用しています。このモードはサポートされていません。」は、情報として表示されるだけです。そのままダイアログを確定して変換を進めてください。BBOXリソースは、バージョン3の動作(ON EVENT CALL、ON SERIAL PORT CALL、セマフォのロック、トランザクションの処理、 Activated/Deactivatedなど)を使用するための互換性チェックボックスでした。

2 - アップグレードの手順 (Copy-Compact-Convert)

- バックアップコピーを作成する。(Copy);
- そのバージョンで入手できる最新の 4D Tools でデータベースを圧縮する。(Compact);
- 新しいバージョンにデータベースを変換する。(Convert)

アップグレードによって露呈したデータベースの不具合に対処する

不具合が起きる理由

アップグレードが原因でデータベースが壊れることはまずありません。むしろ、それまで潜在していた問題が、バージョンアップグレードを契機に表面化するというケースがほとんどです。不具合には、ストラクチャが壊れているために起こるものと、データが壊れているために起こるものがあります。いずれにしても、データが破損したほんとうの原因を突き止めることは非常に困難です。

データの修復

前述の例では、データベースをバージョン6.5に変換することができましたが、その後、読み取れないレコードがあるというアラートが表示される、というのがデータファイルが壊れている場合の代表的な症状です。アラートは、壊れているレコードの数だけ表示されます。すべてのダイアログでOKボタンをクリックした後、4D Toolsによるデータの修復ができるようになります。4D Toolsは、使用中のバージョン(この場合はバージョン6.5)の最新版を使用してください。バージョンが新しいほど性能が良いからです。4D Toolsによる修復がうまくいかないようであれば、変換前のバージョンに戻り、そのバージョンにおける最新版の4D Toolsを使用して修復を試みてみてください。

4D Toolsでデータベースを開く際に次のようなメッセージが表示されるかもしれません:

「データファイルが著しく破損している模様です。"タグによる修復"をおこなう必要がありますが、データがセグメントに分けられている場合は、まず始めに残りのデータセグメントを選択しておく必要があります。」OKをクリックした後、データセグメントが選択できるようになります。

データの修復には次のような方法があります:

- 4D Toolsを使用する。
- データファイルのインデックスを再構築する。
- レコードをロードしてふたたび保存する。
- 新しいデータファイルにデータをインポートする。

4D Toolsを使用する

4D Toolsは4th Dimensionに付属しているユーティリティソフトウェアです。4D Toolsにはデータベースの診断および復元のメニューがあります。軽い破損であれば4D Toolsで修復できるかもしれません。

- 1 - 4D Tools でストラクチャファイルを開く。
- 2 - 破損したデータファイルを選択する。
- 3 - 「管理」タブをクリックする。

データの検査方法には「すべて検査」「レコード検査」「インデックス検査」があります。

「すべて検査」を選択した場合、レコードとインデックスの両方が検査の対象となります。もっとも確実な方法ですが、それだけ時間も長くなります。

定期メンテナンスを実施する場合、あるいはレコードとインデックスのどちらが問題なのかを特定できない場合、このオプションを選択するようにしてください。

検査の結果、何らかの問題が見つかった場合、「修復」タブをクリックしてデータファイルを修復することができます。データの修復方法には「すべて修復」「レコード修復」「インデックス修復」があります。

検出された問題の種類に応じ、修復方法を選択してください。

タグによる修復

4D Toolsの「タグによる修復」を実行する前に他の方法を試みてください。はじめにデータファイルのインデックスを再構築する方法、それで解決しなければ、レコードをロードしてからふたたび保存する方法を試し、それでも問題が解決しなければ、「タグによる修復」を検討するようにしてください。

4D Toolsの修復オプションやデータファイルのインデックス再構築で修復できないデータファイルは、タグによる修復を実行しなければ修復できないのかもしれませんが、このオプションでは、新しいデータファイルが作成されるため、十分なディスク容量があることをあらかじめ確認しておく必要があります。

4th Dimensionのレコードデータは、固定長ではないため、保存されている場所のアドレス情報がアドレステーブルに保存されています。タグによる修復では、各レコードのヘッダ情報(タグ)を元にアドレステーブルの修復が試みられます。タグは、アドレステーブルにおけるレコードの位置を特定するための情報です。

新規データファイルに名前を付けて保存ボタンをクリックすると4D Toolsがタグによる修復を開始します。

4D Toolsのタグによる修復では解決しないほど深刻な場合、もっと強力な方法を探さなくてはなりません。

データファイルのインデックスを再構築する

データファイルのインデックスを再構築するための手順は次のとおりです:

- 1 - ストラクチャファイルのコピーを作成する。
- 2 - コピーを起動して新規データファイルを作成する。
- 3 - すべてのリレートを削除してからすべてのインデックス属性を外す。
- 4 - データベースを閉じる。
- 5 - 元のデータファイルと共にストラクチャを開いてから閉じる。

以上でインデックスのないデータファイルが出来上がります。

- 6 - オリジナルのストラクチャでインデックスのないデータファイルを開く。インデックスが再構築される。

レコードをロードしてからふたたび保存する

データベースを起動することはできても、クエリを実行した途端にクラッシュしてしまい、4D Toolsでデータファイルを検査しても問題が見つからないような場合があるかもしれません。

問題のあるレコードを特定できるもっとも高速で確実な方法は、レコードをひとつずつロードしてからふたたび保存するというものです。壊れたレコードにアクセスすれば、何らかの問題が生じます。テクニカルノート05-09「破損したデータファイルを復旧する」には、すべてのレコードをロードし、問題なく保存できたレコードをログファイルに出力するメソッドが収録されています。メソッドをインポートするには、デザインモードで新規メソッドを作成し、メソッドメニューの「メソッド読み込み」を選択します。

新しいデータファイルにデータをインポートする

データファイルが深刻な損傷を被ってしまい、前述のどの方法をもってしても修復できない場合、最後の手段となるのがデータ全体を新しいデータファイルにエクスポートするというものです。データファイルのサイズによっては、これは非常に時間を要する作業です。エクスポートの方法にも気を配る必要があります。**SEND RECORD**および**RECEIVE RECORD**コマンドでデータをエクスポート/インポートすれば良いように思えるかもしれませんが、そのような方法では、多くの場合、損傷したレコードが複製されるだけです。むしろ標準の書き出し/読み込みダイアログを使用するようにしてください。ファイル形式は、タブ区切りのテキストファイルまたはXML形式を選択します。いずれの操作もユーザモードで実行することができます。ファイルメニューから「書き出し」を選択し、ダイアログが表示されたらエクスポートの対象となるテーブルおよびフィールドを指定してからフォーマットを設定します。全レコードあるいはセレクションを書き出すことができます。最後に「書き出し」ボタンをクリックします。この方法の利点は、それぞれのフィールドにつき、レコードデータがテキストに変換されるため、データの損傷が吸収されることにあります。

アドレステーブルの損傷

どのような症状(インデックスが壊れている、データにアクセスしようとするクラッシュする、など)であるにせよ、変換後にアドレステーブルを修復するにはタグによる修復を実行しなくてはなりません。修復ができない場合、あるいは十分な数のレコードが修復できない場合、今度は変換前のデータベースでタグによる修復を試みてください。いずれにしても、4D Toolsのバージョンが対応しているOSで実行することが推奨されています。タグによる修復で問題が解決しない場合、テクニカルノート4D-200410-29-FR「インデックスのエラーを検出する」を参照してください。

注記: 上位バージョンでファイルを開いてアップグレードすることは、比較的簡単な操作であり、完全に対応していないOSであっても実行できるかもしれません。しかし、タグによる修復のような高度な操作を実行する場合には、完全に対応しているOSを使用しなくてはならない点に注意してください。

ストラクチャの損傷

ストラクチャの損傷がアップグレードに関連して起こることは滅多にありません。データとストラクチャのどちらにも問題があるのかを見極めるには、変換されたデータベースで新規データファイルを作成してみてください。新規データファイルでも問題が起こるようであれば、それはストラクチャの問題です。

4D Toolsのバージョン6.5以降には、ストラクチャを修復するためのオプションが用意されているおり、ストラクチャの問題を解決するための足がかりとなるはずです。4D Toolsの問題解決能力は、バージョンを追うごとに向上しているので、いつでも最新のバージョンを使用するようにしてください。(バージョン2003であれば2003.8、バージョン2004であれば2004.4)

フォームオブジェクトの損傷

フォームやフォームに配置されたオブジェクトの損傷が明らかになる場合がまれにあります。一部のフォームやオブジェクトが表示されなかったり、正しく印刷されなかったりするかもしれません。とりわけ過去のバージョンで作成された後、かなりの期間が経過しているストラクチャでそのような現象が起きるようです。そのような場合、フォームに何らかの変更を加えてから保存すると問題が解決することがあります。フォームを閉じてから再度開き、加えた変更を元に戻せば、フォームが修復できたことになります。あるいは、すべてのオブジェクトを新しいフォームにコピーすれば、やはり同じように修復することができます。特定のオブジェクトが損傷しているようであれば、適当なオブジェクトを元のフォームにコピー&ペーストすることによって、すべてのオブジェクトをリセットすることができます。

- バージョン2004.3の4D Toolsには下記のオプションがあり、通常、前述の手順を踏む必要はありません：
- ストラクチャタブ/ストラクチャ検査&修復/修復ボタン- すべてのデータベースフォームを再生成：



3 – 4D 2004

ストラクチャとデータ: リソースフォークとデータフォークの分離

4D 2004で作成されたデータベースは、Mac OS/Windows共に次のファイルで構成されるようになりました：

.4DB
.RSR
.4DD
.4DR

変換されたMac版のデータベースは、従来のようなストラクチャファイル/データファイルが使用されます。

そのようなデータベースを上記のファイル構成に変換する方法はいくつかあります：

- Tools 2004.3 を使用してデータファイルとストラクチャファイルを圧縮する。
- 4D Transporter を使用して変換する。(Classic 環境)
- Mac OS X のターミナルを使用して変換する。

```
mkdir .../Win_Folder
```

```
cp MyDatabase Win_Folder / MyDatabase.4DB
```

```
cp MyDatabase /..namedfork/rsrc Win_Folder / MyDatabase.RSR
```

```
cp MyDatabase.data Win_Folder / MyDatabase.4DD
```

```
cp MyDatabase.data/..namedfork/rsrc Win_Folder / MyDatabase.4DR
```

新しいアーキテクチャとオペレーションシステム

- 新世代オペレーションシステムとの融和。(Windows XP および Mac OS X)
- ネイティブインタフェースのサポート。
- Mac OS X の Mach-O フォーマットに対応。

2004プロダクトラインは、下記のシステムでは動作しません:

- Mac OS 9
- Mac OS X の Classic 環境
- Windows 98 SE
- Windows Millennium

プラグインフォルダの互換性

- Mach-O (Mach Object-ファイルフォーマット)は、Mac OS X のネイティブ実行ファイルフォーマットです。
- PEF (Preferred Executable Format)は、Mac OS X 以前の Mac OS Power PC システムのネイティブ実行ファイルフォーマットでした。一部の例外を除き、Mac OS X でも実行することができます。
- PEF は、Code Fragment Manager によって実行されるため、CFM と呼ばれる場合もあります。

新しいフォルダ

Extras: 4D Clientに配信されるファイル(リソース、テキスト、設定ファイル、オブジェクトライブラリなど)を保存するフォルダ。ストラクチャと同じフォルダに作成する必要があります。クライアントマシンのコピー先は次のとおりです:

- Mac OS X: Library: ApplicationSupport:4D: データベース名_アドレス:Extras
- Windows 2000/XP: Documents and Settings¥ユーザ名¥Application Data¥4D¥データベース名_アドレス¥Extras

Licenses: アクティベーションによって受信したライセンス、および手動で追加されたライセンスを保存するフォルダ。アクティブな4Dフォルダの中に存在します。

- Mac OS X:Library:Application Support: 4D:Licences
- Windows 2000/XP:Documents and Settings¥All Users¥Application Data¥4D¥Licences

Macros: 作成したマクロファイルを取容するためのフォルダ。

設定フォルダの新しい場所

4th Dimension 2004.3では、設定フォルダ(アクティブな4Dフォルダ)の標準設置場所が変更されました。これにより、All Usersフォルダに対するアクセス権のないシステムでも、4Dが使用できるようになりました。デフォルトの場所は次のとおりです:

	Windows 2000/XP	Mac OS X
4D Client 以外のアプリケーション	Documents and Settings¥All Users¥Application Data¥4D -> All Users フォルダがロックされている場合: Documents and Settings¥ユーザ名¥Application Data¥4D	(起動ボリューム):Library:Application Support:4D -> All Users フォルダがロックされている場合: (カレントユーザフォルダ):Library: Application Support:4D
4D Client	Documents and Settings¥ユーザ名¥Application Data¥4D	(カレントユーザフォルダ):Library:Application Support:4D

4D 2004 よりも前のバージョンでは次のようになっていました:

- 4D Client 以外のアプリケーション:
Windows 2000/XP: Documents and Settings¥All Users¥Applications Data¥4D
Mac OS X: (起動ボリューム) Library:Application Support:4D
- 4D Client:
Windows 2000/XP: Documents and Settings¥ユーザ名¥Applications Data¥4D
Mac OS X: (カレントユーザフォルダ): Library:Application Support:4D

新しいプラグインアーキテクチャ

プラグインはCarbon Mach-OフォーマットあるいはCarbon CFM (Code Fragment Manager)フォーマットのどちらかでなければなりません。(CFMは従来のフォーマット)

Mach-Oフォーマットで作直されたのは、次のプラグインです:

4D View、4D Write、4D Draw、4D for OCI、4D ODBC Pro、4D Open for 4D、4D Pack、4D Internet Commands、4D for MySQL、4D for PostgreSQLおよび4D for ADO。

4D 2004で使用できなくなったのは次のプラグインです: 4D Calc、4D List、4D ODBC (旧ODBCプラグイン)

新しいインストール場所

アクティブな4D Folderにプラグインを置くことはできなくなりました。プラグインをインストールする場所は、データベースストラクチャレベル、アプリケーションレベルのどちらかです。

プラグイン用の新しいフォルダ

新しいフォルダの名前は「PlugIns」です。「.bundle」形式のファイルには、Mac OS/Windows両方のバージョンを含めることができ、Mach-OおよびCFMパッケージはどちらもこのフォルダに置くことができます。ストラクチャレベルに「PlugIns」フォルダと「MAC4DX」「WIN4DX」フォルダを混在させても構いません。Mac OSでは、情報ダイアログの「プラグイン」セクションを使用してプラグインをアプリケーションレベルにインストールすることができます。



この場合、アプリケーションレベルにインストールしたプラグインを非アクティブにすることができます。

クライアント/サーバでプラグインをダウンロードする

4D Client が 4D Server に接続すると、サーバにインストールされたプラグインが自動的にクライアントマシンへ転送されます。ダウンロードされたプラグインは、サーバ側における設定に応じ、クライアントマシンの PlugIns、Mac4DX/Win4DX フォルダのいずれかに置かれます。

注記: 4D Server 2004 では、フォルダの階層構造(フォルダおよびファイル)がクライアントマシン側に再現されます。転送されるのは、クライアントマシンのプラットフォームに関する項目だけです。

注記: 4D Server の内部的な処理のため、Cache、Server および Database というフォルダも作成されます。

クライアントマシンにダウンロードされたフォルダの場所

- PlugIns、Mac4DX/Win4DX フォルダがサーバ側でデータベースストラクチャと同じレベルに置かれている場合:

Mac OS X

{ディスク}:Library:Application Support:4D:データベース名:PlugIns(または Mac4DX/Win4DX)

Windows 2000 Windows XP

{ディスク}:¥Documents and Settings¥ユーザ名¥Application Data¥4D¥データベース名¥PlugIns(または Mac4DX/Win4DX)

- PlugIns フォルダがサーバ側で 4D Server アプリケーションと同じレベルに置かれている場合:

Mac OS X

{ディスク}:Library:Application Support:4D:データベース名:4D:PlugIns

Windows 2000 Windows XP

{ディスク}:¥Documents and Settings¥ユーザ名¥Application Data¥4D¥データベース名¥4D¥PlugIns

読み込みの優先順位

- 同じプラグインがストラクチャレベルとアプリケーションレベルにある場合:
-> ストラクチャレベルにあるプラグインが読み込まれます。
- 同じプラグインが Mac/Win4DX と PlugIns フォルダにある場合:
-> エラーメッセージ(データベースを起動できません。)

ライセンスのアクティベーション

オンラインアクティベーション

4th Dimension 2004のユーザ登録は、基本的にインターネットで行なわれます。

ステップ 1: アカウントを作成するためには、インターネット接続と電子メールアカウントが必要です。アカウントを作成し、後に受け取る通知メールは大切に保管しておきます。

ステップ 2: 製品番号(Product Number)を入力します。

サーバから受信したライセンスファイルが自動的に Licenses フォルダにインストールされます。

万一、自動的にインストールされたファイルが紛失してしまった場合のため、ライセンスファイルのコピーがメールで送られます。

オフラインアクティベーション

ステップ 1: オフラインアクティベーションのダイアログ画面で製品番号とメールアドレスを入力し、シードIDファイル(Reg.txt)を生成します。そのファイルは、インターネットに接続されたマシンにコピーします。

ステップ 2: 4Dの登録Webサイトに接続し、まだ作成していなければ、ここでアカウントを作成します。製品番号を登録しようとする段階で「Reg.txt」ファイルのアップロードが求められますが、ファイルと引き換えにライセンスファイルを受け取ることができます。

ステップ 3: ライセンスファイルを当初のマシンにコピーし、オフラインアクティベーションのダイアログ画面でファイルを選択してインストールを完了します。

緊急アクティベーション

- 緊急アクティベーションができるのは、ひとつの Product Number につき一回です。
- 緊急アクティベーションには、特定の Product Number だけが使用できます。
- 緊急アクティベーションの有効期限は、最長で5日間です。

拡張ライセンスの追加

- ヘルプメニュー: ライセンス更新を選択する。
- コンパイルされたデータベースであれば **CHANGE LICENSES** コマンドを使用する方法もある。

拡張ライセンスが最初から組み込まれた製品(Web Serverライセンス付の4D Serverなど)は、Product Numberの中に拡張ライセンスが含まれているので、ライセンスの更新をする必要はありません。ライセンスファイルの内容は、ブラウザで開いて確認することができます。

4 – 変換されたデータベース: 互換性

互換性は、新しいメカニズムではなく従来の方式を利用したい人のために残されているオプションです。いうまでもなく、互換モードはいろいろな面で劣っており、使用することは推奨されていません。一部の互換性オプションは、変換されたデータベースでなければ選択できないようになっています。

ストラクチャの互換性

- 「V3.x.x の Startup プロシージャ方式を使う」: バージョン 6 以前の Startup プロシージャコールを有効にするオプションです。
Tip: オプションを外し、Startup の内容を On Startup データベースメソッドにコピーしてください。
- 「V3.x.x のファイルプロシージャ方式を使う」: 入力フォームでフィールドにデータを入力するとき、あるいはボタンをクリックしたときのファイルプロシージャを有効にするオプションです。
Tip: オプションを外し、トリガを使用してください。
- 「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**ではデフォルトで有効、**新しく作成されたデータベース**ではデフォルトで無効のオプションです。
- 「テキスト描画を 6.8 互換にする」: Mac OS X における 100% Quartz 描画を無効にするオプションです。バージョン 6.8 ではフィールド、テキスト変数、プラグインが QuickDraw モードで描画されていました (解析、文字の配置など)。Quartz で画面が描画されるのは、Mac OS X 10.2 以降でプラットフォームプロパティが「Mac テーマ」の場合です。(印刷は無関係)
- 「ラジオボタンを名前でグループ化する」: **変換されたデータベース**ではデフォルトで有効のオプションです。新しく作成されたデータベースでは、ラジオボタンを変数名に関係なくグループ化すれば、グループラジオボタンとして動作します。
- **PRINT SELECTION** 中、レコード毎にフォームをリロードする: **変換されたデータベース**ではデフォルトで有効のオプションです。過去のバージョンでは、レコード毎にフォームがリロードされていたため、**On Printing Detail** フォームイベントで動的に変更されたオブジェクトのプロパティを印刷に反映させることができました。4th Dimension 2004 ではパフォーマンスを向上させるため、リロードメカニズムが廃止されました。オブジェクトプロパティは **On Display Detail** フォームイベントで再設定してください。変換されたデータベースでは、このオプションで過去のバージョンにおける動作を再現することができます。
新しく作成されたデータベースでは、常に新しいメカニズムが採用されます。

- 「データ入力時に自動トランザクションを使用する」: 変換されたデータベースに存在し、デフォルトで無効のオプションです。有効の場合、サブフォームから入力フォームが開かれたときにトランザクションが自動的に開始されます。4D First から 4D へ移行したユーザのためのオプションで、バージョン 2003 の時点で使用中止が推奨されていました。

Web 互換性

- 「ブラケットの代わりに 4DVAR コメントを使用する」: 過去のバージョンでは、次のシンタックスが使用されていました: [MyVar]。通常は次のシンタックスに置換してください: <!--4DVAR MyVar-->
- 「新しいコンテキスト参照モードを使用」: ページの各要素ごとのコンテキスト参照番号をブラウザに送信しないためのオプションです。
- 「未知の URL から "/" を取り除く」: 過去のバージョンでは、たとえば <http://127.0.0.1/Contacts/Add> で「Contacts/Add」が \$1 に返されていました。このオプションを使用しない限り、\$1 の冒頭は "/" になります。
- 「Keep-Alive 接続を使用する」: 過去のバージョンでは、たとえば HTML ページに画像が 10 枚ある場合、HTTP 1.0 が 11 接続発生していました。このオプションを使用すれば、1 接続で済みます。

メニューバー

旧バージョンの編集メニューを使用(4D 2003のオプション)

過去のバージョンで作成され、4D 2003にアップグレードされたデータベースでは、「旧バージョンの編集メニューを使用」というオプションが利用できました。4D 2004では、同じオプションの名称が「v 6.8」になりました。バージョン2003で作成されたデータベースには当てはまらないからです。アップグレードの前から存在するメニューバーは、このオプションがデフォルトで有効にされており、4th Dimensionによって自動的に編集メニューが追加されます。そのような場合、過去のバージョンにおける動作は保持されますが、編集メニューを変更することはできなくなります。

イベントを発生させない

「イベントを発生させない」オプションが有効にされている場合、選択された標準アクションを実行する前にカーソルが置かれているオブジェクトの確定イベントを省略します。基本的に編集コマンドを設定されたメニューのためのオプションです。通常、メニューコマンドまたはショートカットで標準アクションを実行する前に入力中の内容が確定するため、**On Data Change** フォームイベントが発生します。コピーやペーストのような編集コマンドの場合、このタイミングで**On Data Change**が発生することは望ましくないかもしれません。「イベントを発生させない」オプションを有効にすることでこの問題を回避することができます。

プラットフォーム (環境設定およびフォームプロパティ)

変換されたデータベースの環境設定ダイアログには、プラットフォームに関する設定項目があります。過去のバージョンでは、データベース、フォーム、オブジェクトレベルで特定のプラットフォームアピランスを強制することができました。このメカニズムは互換性のために残されています。

4th Dimension 2004以降で作成されたデータベースの場合、アピランスは実行中のプラットフォームに合わせて自動的に選択され、フォームオブジェクトはネイティブの方式で描画されます。アピランスは、フォームレベルかオブジェクトレベルのどちらかで設定し、選択肢はシステムまたは印刷、オブジェクトレベルであればフォームから継承の3種類です。

- システム**: 上位レベルの設定を無視し、プラットフォームネイティブのアピランスを採用します。
- 印刷**: 上位レベルの設定を無視し、印刷用のアピランスを採用します。
- フォームから継承**: 上位レベル、つまりフォームの設定を継承します。

過去のバージョンから変換されたデータベースは、自動、Mac OS 7、Windows 3.11、NT 3.51、Windows 95/98/2000、NT 4、Mac OS 9、およびMacテーマのアピランスが選択できます。これらのオプションは、4th Dimension がサポートしていた過去のOSインタフェースを再現するためのものです。特別な理由がなければ使用しないようにしてください。**Get platform interface**および**SET PLATFORM INTERFACE**コマンドは、変換されたデータベースでしか動作しません。4D 2004で作成されたデータベースは、プラットフォームインタフェースがシステムに依存しているため、どちらのコマンドも無視されるようになっています。

変換されたデータベースの場合、**環境設定の互換性ページ**(「アプリケーション」テーマ)でプラットフォームを「システム」に変更することが強く推奨されています。そのようにすれば、自動選択のオプションは無効になり、新しいメカニズムが適用されるようになります。ただしフォームにはそれぞれ独自のプロパティがあるので、設定が反映されるためには、フォームのプラットフォームプロパティがデータベースから継承などに設定されている必要があります。

- **データベースから継承:** フォームのプラットフォームインタフェースは環境設定から継承されます。

フォームウィンドウのサイズ

ウィンドウサイズ: 制約付き(4D 2004.2以降に変換されたデータベース)

選択しなかった場合: 過去のバージョンにおける**サイズ変更可**が有効にされていないときの動作。

選択した場合: バージョン2004の標準的な動作。

4D 2004.0: 常にフォームサイズを自動計算

変換されたデータベースは、フォームの「サイズを決めるもの」を設定した場合、常にフォームサイズを自動計算というオプションがプロパティリストに追加されます。

4th Dimensionの過去のバージョンでは、フォームを表示するときに一度だけフォームサイズが計算されました。その結果、フォームの**サイズを決めるもの**を**MOVE OBJECT**で移動したとしても、フォームサイズは変わりませんでした。「常にフォームサイズを自動計算」オプションは、フォームサイズを動的に設定するためのオプションです。(変換されたデータベースは、デフォルトでこのオプションが選択されていません。)

4D 2004.2以降: 制約つき

バージョン2003から2004.2に変換されたデータベース: このオプションは、過去のバージョンにおける「**サイズ変更可**」が有効にされていないときの動作(ユーザはフォームをリサイズできるが、オブジェクトは移動もリサイズもされない)を再現するために使用することができます。その場合、フォームのサイズ設定、最大/最小の幅/高さはすべて無視されます。

この互換性オプションは、4th Dimensionの過去のバージョンにおける動作を再現するためのものです。特別な理由がなければ使用しないようにしてください。

なお、フッタに配置されたオブジェクトの意図しないリサイズは、このオプションで防ぐことができます。バージョン2004.0で作成されたデータベースは、バージョン2004の標準的な動作になります。

システムハイライトカラー

変換されたデータベースは、フォームのプロパティリストに「システムハイライトカラー」というオプションがあり、有効にした場合、選択された行はシステムの設定でハイライト表示されるようになります(システムハイライトは背景ピクチャに対しては効果がありません)。無効の場合、選択された行は反転色でハイライトされます。

ASCII テーブル編集

過去のバージョンでは、ユーザモードの特別メニューにあった「ASCII 入力/出力テーブル編集」は、環境設定のオプションページ(「Web」テーマ)に移り、「入力/出力フィルタを編集」という名称になりました。

プラグインの対応

4D Write

バージョン 6.x.xからバージョン2003または2004にアップグレードした場合、保存されていた4D Writeドキュメントには無効な印刷設定(A4 100%のはずがUSフォーマット144%になっているなど)が含まれていることとなります。この場合、パーセント値を落とすだけでは問題が解決しません。原因は、「Print record」リソースがMac OS 9とMac OS Xでは異なることにあります。

以下はそのようなドキュメントを4D 2004で修正するためのメソッドです:

C_BLOB(MyBlob)

MyArea:=**WR New offscreen area**

　`新しいデフォルト設定値を BLOB に保存。

MyBlob:=**WR Print settings to BLOB** (MyArea)

　`問題のドキュメントを開く。

WR OPEN DOCUMENT (MyArea)

bOK:=OK

While (bOK=1)

　`設定を置換。

WR BLOB TO PRINT SETTINGS (MyArea; MyBlob;1)

WR SAVE DOCUMENT (MyArea)

　`繰り返し。

WR OPEN DOCUMENT (MyArea)

　bOK:=OK

End while

WR DELETE OFFSCREEN AREA (MyArea)

警告: 4D Writeのドキュメントフォーマットはバージョン2004.2で変更されました。2004.2以降で作成または保存されたドキュメントを2004.2よりも前のバージョンで開くことはできません。

5 - 統合されたアプリケーション

統合された4D Insiderの機能

エクスプローラのフォルダと4D Insiderのグループ

4D 2004では、デザインモードでフォルダやサブフォルダが作成できるようになりました(エクスプローラのホームページ)。作成されたフォルダには、テーブル、フォーム、プロジェクトメソッドを整理して収容することができます。

メソッドエディタでは、作成されたフォルダが階層リストとして表示されます。

エクスペローラのフォルダは、4D Insiderのグループとしてデータベース間で移動することもできます。

4D Insider グループ	4D フォルダ
テーブル	テーブル
フォーム	フォーム
プロジェクトメソッド	プロジェクトメソッド
メニューバー	非表示
リスト	非表示
スタイルシート	非表示
フォーマットとフィルタ	非表示
ヘルプ Tips	非表示
データベースメソッド	非表示
STR# リソース	非表示

4D オブジェクトライブラリと4D Insiderのライブラリ

4D Insider ライブラリ	オブジェクトライブラリ
拡張子.4il の外部ファイル	拡張子.4il の外部ファイル
データベースを構成するあらゆるオブジェクト	プロパティを持つフォームオブジェクトとそれに関連するアイテム(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、ヘルプ Tips、STR# リソース、オブジェクトメソッド)

統合された4D Customizer Plusの機能

- スクリプトマネージャ
- 印刷の進捗状況
- スプラッシュスクリーン
- キーボードショートカット
- WEDD リソース
- 4D Server ポート番号

以上の項目は 4D の環境設定ダイアログに組み込まれました。

スクリプトマネージャ

4D 環境設定

TRIC リソース

- 4th Dimension: TRIC リソース、存在しなければ TRI#リソースを使用。
- システム: システムによるクエリおよび並び替え(TRIC リソースを無視)。
- German for v2.2 過去のドイツ語バージョンで使用されていた方式。
- Turkish (トルコ語システムのバグを回避するための処置)。
- 日付による計算(イランの言語 Farsi)

言語設定には、欧米語(Roman)のシステムとスクリプトマネージャによるシステムがあります。

4D 2004では、非ローマ文字のサポートが強化され、同一フォームにおける言語が異なるデータの入力および表示に対応できるようになりました。(例: ハンガリー語、チェコ語、アラビア語をひとつのフォームの異なるフィールドや変数で使用) (プロパティリストのキーボードレイアウトで設定)

Mac OS X: オペレーションシステムで利用できる15言語(英語、日本語、フランス語、ドイツ語、スペイン語、イタリア語、オランダ語、スウェーデン語、デンマーク語、ノルウェー語、フィンランド語、繁体字中国語、簡体字中国語、韓国語およびポルトガル語)が使用できます。

Windows: Microsoft 社より販売されている多言語パックMUI (Multilingual User Interface) (Professional Version)が使用できます。

WEDDリソース

WEDDリソースは、ストラクチャを特定のデータファイルに結びつけるために使用されます。使用すれば、そのストラクチャと同じWEDD署名を有するデータファイルだけが開けるようになります。
4D 2004では、環境設定(データベース>データ管理)でストラクチャファイルとデータファイルのWEDD署名を設定できるようになりました。
WEDD署名を取り除くには、環境設定のWEDD署名の値を削除します。

キーボードショートカット

ユーザモード	2003(Customizer Plus)	アップグレード	2004(環境設定)
入力フォーム確定	Alt/option+Enter		Enter
入力フォームキャンセル	Alt/option+Esc		Esc
サブレコードの追加	Enter		CTRL/command+SHIFT+/ /

4D Server: ポート番号を変更する

ポート番号は、環境設定(クライアント-サーバ/設定)で変更できるようになりました。

スタックサイズ

4D Customizer Plus では、4D のさまざまなスタックサイズも設定することができました。そのような設定はもはや不要であり、設定項目から取り除かれました。New process のようなコマンドのスタックサイズ設定は、互換性のために残されています。

システムおよびコマンド名で使用される言語

アプリケーションの言語は次の方法で変更することができます:

Macintosh: 4D アプリケーションパッケージの Contents:Resources フォルダにある言語フォルダ、つまり日本語版であれば「Japanese.lproj」と同じ場所にフランス語であれば「French.lproj」フォルダを追加します。それからアプリケーションの「情報を見る」ダイアログの言語セクションで希望の言語を選択します。

Windows: フランス語版の 4D から「4D.rsr」ファイルをコピーし、英語版の「4D.rsr」を置換すれば、アプリケーションがフランス語版になります。

プラグインに統合された4D Customizerの機能

すでに統合されている4D Drawのカスタマイズ項目

4D Drawのコマンドに統合された4D Customizer Plusの項目:

テンプレート

DR SET GLOBAL PREFERENCES

オブジェクトとして張り付け

自動保存のアラートなし

DR SET PREFERENCES

サポートされなくなった項目: 外部ウインドウの接尾辞(DR)

追加されたパラメータ

DR SET ENTERABLE(area; mode{;buttonMode})buttonMode 整数
 0: 高さ 150 ポイント/幅 300 ポイントに満たなければエリアをボタン表示する。
 1: ボタン表示しない。 **DR SET GLOBAL PREFERENCES**(read;
 write{;selection{;grid{;horLock{;verLock}}}))
 selection -> 整数
 0 変更しない。
 1 接触オブジェクトを選択する。
 2 内包オブジェクトを選択する。
 grid -> 整数
 0 変更しない。
 1 グリッドを非アクティブにする。
 2 グリッドに合わせる。
 horLock -> 文字列
 横方向ロックのキー(デフォルト: クリック + H)
 verLock -> String
 縦方向ロックのキー(デフォルト: クリック + V)

4D Backupの統合と新しいバックアップメカニズム

新しくなったこと

- 刷新されたアルゴリズム。
- 自動バックアップと自動復元 (ストラクチャ、データ、添付ファイルを含む)。
- 4D、4D Server、4D Runtime Volume License によるパラメータの設定。
- 自動でログファイルを統合。
- ロールバックの導入。
- 圧縮およびアーカイブセキュリティの強化。
- XML形式のバックアップ設定。
- さまざまなバックアップ方法を提供:
 - デザインモード > 環境設定 > バックアップ
 - ユーザモード > ファイルメニュー > バックアップ
 - **BACKUP** コマンド: 開始時 **On Backup Startup**、終了時 **On Backup Shutdown** データベースメソッドを実行。

新データベースメソッドおよび新コマンド

On Backup Startup database method

On Backup Shutdown database method

BACKUP

RESTORE

GET BACKUP INFORMATION(selector;info1;info2)

GET RESTORE INFORMATION(selector;info1;info2)

生成されるファイル

データベース名[XXX1-0002].4BK: 1 番目のバックアップの 2 番目のセグメント。
 ログファイル名[XX12].4BL: 13 番目のログファイル(0 から開始するため)。

その他

- 圧縮率: 高い場合、所要時間は長くなるが、ディスクスペースが節約できる。
- インターレース率: 高い場合、所要時間は長くなるが、セキュリティが向上。(セクターが隣接しない)
- 冗長率: 高い場合、所要時間は長くなるが、セキュリティが向上する。(大きなファイル)
- 新規データベース作成時に自動バックアップを有効にするオプション。
- ようこそ画面に復元オプション。

ログファイル

- バックアップを実行するたびに新規ログファイルを作成。
- あらゆる追加、更新、削除、トランザクションを記録。
- ログファイルをチェック: ユーザモード> ファイル>ログファイルをチェック (4D Mono/4D Server) - アクション、テーブル、ユーザ、日付と時刻、フィールドの値をブラウズ。
- ロールバック: 選択された行から後の操作を取り消し。

自動復元

データベースを開くときに自己診断を実行:

- データベースが壊れていたら、最新のバックアップから復元する。
- データベースが完全でない場合、最新のログを統合する。
- 自動復元の後、データベースを開始する。(バックアップ環境設定のオプション)

サービスとして登録された4D Serverを自動的に再起動する

手動で復元する

- データベースを開くときに復元タブを選択。(4D Mono/4D Server)
- **RESTORE** コマンド。
- バックアップファイルをアプリケーションのアイコンにドロップ。(すべての 4D アプリケーション)

バックアップログとXML設定ファイル

Backup log.txt: バックアップのパス名、サイズ、アーカイブの内容など。

「Backup.xml」設定ファイルの保存場所: データベースフォルダ:Preferences:Backup:

統合された4D ODBC

ハイレベル ODBC コマンドを実装。4D ODBC 2003 プラグインと 4D ODBC Pro 2004 プラグインの間に互換性はありません。一方、基本的な ODBC は 4D 2004 の標準コマンドに統合されました。

- すべての 4D 2004 製品が対象。
- 追加ライセンスは不要。
- ODBC データソースに接続してインポート & エクスポートを実行。
- ODBC データソースには SQL でアクセス。
- ODBC 3.0 API
- Mac OS X ネイティブ。
- ODBC マネージャは Windows および Mac OS X に標準で付属。
- Windows: ODBC ドライバが標準で複数利用可能。
- Mac OS X: 標準ドライバなし。サードパーティ社製品:<http://www.openlinksw.com/> API ドキュメント:
- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hm/odbcodbc_api_reference.asp
- ODBC 2003 コマンドを 2004 コマンドで置換するためのコンポーネント「ODBC Framework」(製品 CD またはダウンロードでインストールされた ODBC Pro プラグインに付属)。

4D 2004で追加されたプラグイン

- 4D for MySQL
- 4D for PostgreSQL
- 4D for ADO

ユーザ&グループの管理

BLOB TO USERS (users)

USERS TO BLOB (users)

管理者だけがこれらのコマンドを実行できます。

ツールボックスでも似たような操作ができますが、その場合は書き出し先がファイルに限定されます。

互換性に関する注記: 4th Dimension 2003で作成されたユーザアンドグループファイル(拡張子.4UG)は、上記の方法で4th Dimension 2004に読み込むことができます。

6 – 動作面で新しくなったこと

フォーミュラエディタ

フォーミュラエディタにプロジェクトメソッドは表示されなくなりました。カレントセッションにおけるプロジェクトメソッドの使用は **SET ALLOWED METHODS** コマンドで明示的に許可しなくてはなりません:

SET ALLOWED METHODS (methodsArray)

GET ALLOWED METHODS (methodsArray)

EDIT FORMULA (table; formula)

例:

ARRAY STRING (15;methodsArr;2)

methodsArr {1}:="Formula@"

methodsArr {2}:="Grand total"

SET ALLOWED METHODS (methodsArray)

\$myFormula:=""

EDIT FORMULA ([Salaries];\$myFormula)

If (OK=1)

APPLY TO SELECTION ([Salaries];EXECUTE(\$myFormula))

End if

ユーザモード、カスタムモード、サブフォームで表示されたリストフォームの動作を統一: すべてのリストフォームに適用される選択モード

- なし

UserSet は作成されません。 **On Selection Change** フォームイベントは発生しません。

- 単一

UserSet は作成されません。

選択されたレコードがアンロードされたカレントレコードになります。

UP および DOWN 矢印キーで選択を変更することができます。

CTRL/command+クリックで選択/非選択を切り替えることができます。

On Selection Change はカレントレコードが変わるたびに発生します。

- 複数

UserSet が作成されます。(警告: サブフォームの場合、UserSet は **GET HIGHLIGHTED RECORDS** コマンドで取得することになります。 - 後述を参照してください。)

クリックでカレントレコードが変わることはありません。

UP および DOWN 矢印キーが使用できます。

CTRL/command+クリックまたは SHIFT+クリックで複数のレコードが選択できます。

On Selection Change イベントは選択されたレコードが変わるために発生します。DISPLAY SELECTION コマンドなどの *enterList* パラメータが **True** に設定されていた場合、あるいはサブフォームのリスト更新可プロパティが選択されている場合、レコードの追加と更新はリストフォームで実行できます。そうでない場合、詳細フォームが使用されます。

リストフォーム関連

- 自動的に追加される終了ボタンの廃止。
- ヘッダ、フッタ、ブレイク、詳細エリアでアクティブオブジェクトをサポート。
- 標準アクション「サブレコード追加/更新」が通常のリストフォームに対応。

リストフォーム関連の新イベント

- **On Load Record:** リスト更新モードに突入するとき。
- **On Selection Change:** リストフォーム、サブフォーム、リストボックスでカレントレコード、カレントセレクション、選択対象が変わったとき。

リスト関連の新コマンド

GET HIGHLIGHTED RECORDS (*{table};setName*)

setName: ユーザが作成したセットのコピーを受け取るセット名を渡します。

テーブル名が指定できるのため、サブフォームの *UserSet* が受け取れるようになりました。

セットはローカル/クライアント、プロセス、インタープロセスセットのいずれかを使用します。

SCROLL LINES (*{*};object|table;position;{*}*)

- * サブフォームの場合はオブジェクト名を使用。(文字列)

- * 指定された項目が最上行になるようにスクロールを実行します。

サブフォーム、**MODIFY SELECTION** または **DISPLAY SELECTION** コマンドで表示されたリストフォームにおいて使用することができます。

position パラメータはオプション; 指定しなかった場合、選択された最初のレコードが一行目になります。

リストフォーム関連

GOTO SELECTED RECORD (*{table};record*)

record パラメータに 0 を渡せば、リストフォームの選択レコードがすべて選択解除できるようになりました。

HIGHLIGHT RECORDS (*{table}; setName;{*}*)

table パラメータにサブフォームのテーブル名が渡せるようになりました。

SCROLL LINES コマンドで制御する場合のために、自動スクロールをキャンセルできるようになりました。

リスト更新関連

- すべてのタイプのリストフォームでリスト更新ができるようになりました。

DISPLAY SELECTION (*{table};selectMode;enterList;{*};{*}*)

MODIFY SELECTION (*{table};selectMode;enterList;{*};{*}*)

selectMode: 0 ない、1 単一、2 複数;

enterList: True または False (デフォルトで False)

サブフォーム関連

- 水平スクロールバー
- 「行をダブルクリック」管理
 - なにもしない
 - レコード更新
 - レコード表示
- 「空行をダブルクリック」の管理 (変換されたデータベース)
 - なにもしない
 - レコード追加(リスト更新または詳細フォーム使用)
- サブフォームオブジェクトはフォーカス可
- 点滅アイコンの廃止

過去のバージョンと4D 2004のサブフォームプロパティの比較:

4D 過去のバージョン		4D バージョン 2004	
入力可	ダブルクリック可	フォーカス可 リスト更新 選択モード 行をダブルクリック 空行をダブルクリック	有効 有効 なし レコード更新 レコード追加
入力可	ダブルクリック不可	フォーカス可 リスト更新 選択モード 行をダブルクリック 空行をダブルクリック	有効 有効 なし 何もしない 何もしない
選択可	ダブルクリック可	フォーカス可 リスト更新 選択モード 行をダブルクリック 空行をダブルクリック	無効 無効 単一 レコード表示 何もしない
選択可	ダブルクリック不可	フォーカス可 リスト更新 選択モード 行をダブルクリック 空行をダブルクリック	無効 無効 単一 何もしない 何もしない
選択不可		フォーカス可 リスト更新 選択モード 行をダブルクリック 空行をダブルクリック	無効 無効 なし 何もしない 何もしない

「空行をダブルクリック」は、変換されたデータベースに存在する互換性のためのプロパティです。現行のシステム(OS)インタフェースガイドラインにはそぐわないため、あまり推奨されていません。

- ダブルクリックできるような空行が表示されていなくても、次の方法でレコードが追加できます:
 - メインフォームに配置されたボタンをクリックする。
 - サブフォームのヘッダまたはフッタに配置されたボタンをクリックする。
 - メインフォームのフッタに配置されたボタンをクリックする。
- カレントレコードが自動的にロードされることはなくなりました:

サブフォームの再描画およびフォームオブジェクトのフォーカス取得によるフォームの再描画は、いずれもカレントレコードのアンロードにつながります。ただし、カレントレコードポインタはアンロードされたカレントレコードを指しています。

対応: **LOAD RECORD** コマンドを使用してください。

- **On Display Detail** イベントは空の行でも発生するようになりました。結果、これまでは表示されなかった変数がリストの最下部まで連続して表示されるという現象が起こる場合があるかもしれません。

コードを少し追加することによって対応することができます:

If (Is record loaded([Table]))

...

End if

または

```
$n:=Displayed line number
$p:=Records in selection([Clients])
If ($n>$p)
  SET VISIBLE (*;"Popup@";False)
Else
  SET VISIBLE (*;"Popup@";True)
End if
```

- ヘッダおよびブレイクエリアのオブジェクトが表示されるようになりました。
- 空の行が表示されるようになりました。
- 選択行/非選択行のオブジェクトをプロパティで非表示にできるようになりました。
- ヘッダおよびブレイクエリアのアクティブオブジェクトが動作するようになりました。
- フォームイベントの実行順序:

バージョン2003以前:

- トリガ
- **On Display Detail** フォームメソッド

4D 2004 (変換または作成されたデータベース) – **MODIFY SELECTION**または **DISPLAY SELECTION**で表示されたフォームと同じ動作。

- **On Header** オブジェクトメソッド
- **On Header** フォームメソッド
- **On Display Detail** オブジェクトメソッド
- **On Display Detail** フォームメソッド

7 – 新しいキーボードショートカット

CTRL/ command +,: エクスプローラを表示する
CTRL/ command +;: ツールボックスを表示する
CTRL/ command +P: メソッドを表示する
CTRL/ command +L: フォームを表示する
CTRL/ command +*: コンパイラを表示する

変更されたキーボードショートカット

CTRL+クリック/command+クリックは、グループ化されたオブジェクトの単独選択になりました。以前の操作は、コンテキストメニュー「同種のオブジェクトを選択」で実行することができます。

CTRLまたは **command+スペース** (テーブルリストを表示する)は、次のように変更されました:

レコード>テーブルリストあるいは

CTRL/command+0 (ゼロ)

CTRL+W または **command+W** は次のように変更されました:

ファイル>ディスクに保存あるいは

ALT CTRL/OPTION command+SHIFT+S

CTRL +クリック (選択したリスト項目の編集)は、次のように変更されました:

ALT/option+クリック

追加されたキーボードショートカット(デザインモード)

ENTER テキストオブジェクトに対するデータ入力を確定します。
ALT+クリック ページ 0 に移動してページ 0 のオブジェクトを選択します。

カスタムモードからユーザモードに切り替える方法

Macintosh: 2003.7 以降、変更なし。ショートカットは引き続きスプラッシュスクリーンで option+F です。
Windows: 従来のショートカット Alt+F4 はデータベースの終了(標準の操作)となりました。
ユーザモードからカスタムモードに切り替えるには、次の操作をしてください:

- 4D 2003.7 以降: Alt+Shift+右クリック - 次のユーザアクションでデバッグが起動。
- 4D 2004: Alt+Shift+右クリック - ユーザモードに移動。

ftp://ftp.4d.com/aci_product_reference_library/4d_product_documentation/pdf_docs_by_4d_product_a-z/4d/4d_2004/4D_2004_Keyboard_Shortcuts.pdf

8 – 4D 2004以降で新しくなったこと

- ローカル変数に対するポインタの使用(2004.1);
- 新しいようこそダイアログ画面と「お気に入り」管理の追加(2004.1);
- 「読み込みのみモードでストラクチャファイルを開く際の警告を表示しない」オプション(環境設定「アプリケーション」テーマのアクセスページ) (2004.1);
- Extras フォルダを経由した 4D Server から 4D Client に対するファイルやフォルダの配信(2004.1);
- サブフォームのプロパティで Backspace/Delete キーによるレコードの「削除を許可」(2004.1);
- メソッドエディタのタイプahead選択メニューで「(」または「;」キー入力に対応(2004.1);
- ライセンス更新メニューまたは **CHANGE LICENSES** コマンドで表示される新しいライセンス更新ダイアログ画面(2004.1);
- SAX コマンドで XML を開く場合は Read Only が必須に変更(2004.1);
- **SET DICTIONARY** コマンドで使用できるスペル辞書に各言語のバリエーションを追加(2004.1);
- データフォークリソースの作成とオープン(2004.1);
- Mac OS 10.4 Tiger 「certified」級の対応(2004.3);
- 環境設定ファイルの標準設置場所(4D フォルダ)の変更(2004.2);
- 「必要に応じて新しいデータセグメントを作成する」オプションの追加(2004.2);
- 「4D Server と 4D Runtime (Macintosh)でキャッシュを物理 RAM に置く」(アプリケーションの実行速度を落としかねない仮想メモリへの転送を許さない)(2004.2);
- フォームエディタにフォームをテストするためのボタンを追加 (2004.2);
- ウィンドウサイズ「制約つき」プロパティ(4D 2004 に変換されたデータベース)(2004.2);
選択しなかった場合、ユーザはウィンドウをリサイズできるがオブジェクトは固定される(過去のバージョンにおける「サイズ変更可」を再現する動作)
選択した場合、新バージョンにおける標準の動作;
- リストボックスの列同士または他のオブジェクトとの間でドラッグ&ドロップ(2004.2);
- メニュー項目のショートカットに Shift キーを設定(2004.2);
- 非ローマ文字のサポート: アラビア語、繁体字中国語、簡体字中国語、クロアチア語、キリル語、ギリシャ語、ヘブライ語、アイスランド語、日本語、韓国語、ルーマニア語、タイ語、ベトナム語。以上はデータの入力と表示、およびデータ処理(並び替え)に対応: 欧米語(Roman)システムあるいはスク립トマネージャシステムのいずれかを使用(2004.2);
4D 2004.2: 同一フォームにおける言語が異なるデータの入力および表示に対応(例: ハンガリー語、チェコ語、アラビア語をひとつのフォームの異なるフィールドや変数で使用) (プロパティリスト: キーボードレイアウトによる設定);
- Mac OS X: オペレーションシステムで利用できる 15 言語(英語、日本語、フランス語、ドイツ語、スペイン語、イタリア語、オランダ語、スウェーデン語、デンマーク語、ノルウェー語、フィンランド語、繁体字中国語、簡体字中国語、韓国語およびポルトガル語)を使用;
- Windows: Microsoft 社より販売されている多言語パック MUI (Multilingual User Interface) (Professional Version)を使用;

ランゲージで新しくなったこと

メソッドエディタ:

選択されたプロジェクトメソッドを開くためのショートカットが追加されました(2004.3):

Alt+ダブルクリック(Windows)または option+ダブルクリック (Macintosh)

新規プロセスのスタックサイズ:

新規プロセスには 64 KB のスタックを配分することが推奨されています。これまでは 32K を超えるスタックが自動的に配分されていましたが、バージョン 2004.3 以降、設定されたスタック値が厳密に適用されます。

追加されたコマンド

SHOW ON DISK (pathName { ;*})

pathName で指定されたフォルダあるいは指定されたフォルダの内容を表示する(2004.1)。

DOM Get parent XML element (2004.2)

DOM Get previous sibling XML element (2004.2)

DOM Get last child XML element (2004.2)

変更されたコマンド

Drop position: リストボックスの列番号を調べるためのパタメータが追加されました(2004.2)。

PLATFORM PROPERTIES: システム言語を調べるための *language* パラメータが追加されました(2004.2)。

SET DATABASE PARAMETER、**Get database parameter:** セレクタ 32 Real Display Precision 実数の切り捨てられる末尾桁数(2004.2)。

SET FIELD TITLES、**SET TABLE TITLES:** *(フォーミュラエディタでバーチャルストラクチャ名を使用。)

PRINTERS LIST: Mac OS でカスタムプリンタ名を使用するためのパタメータが追加されました(2004.2)。

SET DATABASE PARAMETER ({table;}Debug Log Recording; value)セレクタ 34 Debug Log Recording。値: 0: 記録しない; 1: 記録する; 2: 詳細モードで記録する。

「4DdebugLog.txt」という名前のファイルを出力(2004.3)。(警告: すぐに肥大化するので注意が必要です。)

SET PRINT OPTION (Hide printing progress option; value) (2004.3)

GET PRINT OPTION (Hide printing progress option; value) (2004.3)

値: 0: 表示する 1: 表示しない (印刷の進捗状況に関わる環境設定を補完。)

名称が変更されたコマンド

DOM Get first XML element 新名称 **DOM Get first child XML element** (2004.2)

DOM Get next XML element 新名称 **DOM Get next sibling XML element** (2004.2)

SHOW LISTBOX SCROLLBAR 新名称 **SET SCROLLBAR VISIBLE** (2004.3)

リストボックスコマンドの対象オブジェクトがサブフォームとスクロールエリアにも拡大されました。

追加された定数

Get 4D folder コマンド *folder* パラメータの定数 (2004.1): Active 4D Folder、Extras Folder、Licenses Folder および 4D Client Database Folder。

- **SET ENVIRONMENT VARIABLE** コマンドの新しい定数(2004.1):
_4D_OPTION_CURRENT_DIRECTORY (Macintosh: 相対パスを使用する)
_4D_OPTION_HIDE_CONSOLE (Windows: コンソール画面を表示しない)
- **Get database parameter** 関数および **SET DATABASE PARAMETER** コマンドの追加パラメータ:
Get database parameter ([table];Table Sequence Number) (2004.1)

- **SET DATABASE PARAMETER** ([table]; Table Sequence Number;value) (2004.1)
Sequence number が次に返す数値を設定。
Get database parameter ([table]; 4D Server Log Recording) (2004.1)
Get database parameter ([table]; Web Log Recording) (2004.1)
Get database parameter ([table]; Client Web Log Recording) (2004.1)
SET DATABASE PARAMETER ([table]; 4D Server Log Recording) (2004.1)
SET DATABASE PARAMETER ([table]; Web Log Recording) (2004.1)
SET DATABASE PARAMETER ([table]; Client Web Log Recording) (2004.1)
4D Server、Web または 4D Client Web リクエストのログを記録。
- BLOB で SOAP を送るために Is Text 定数(Is BLOB ではない)が使用できるようになりました(2004.1):
SOAP DECLARATION (blob, Is Text;input output)
上記コマンドで BLOB による 32KB 超の SOAP が返せるようになりました。

追加されたフォームイベント

On After Edit: ペースト、カット、削除、取り消し、やり直し、ドロップおよびタイプ入力を含む操作、または操作を擬似的に再現するコマンドによって入力可エリアが編集されたとき(2004.2)。
(より限定的な **On After Keystroke** に代わるフォームイベント)

バックアップで新しくなったこと

新しいバックアップコマンド:

New log file

-> テキスト: 閉じられたログファイルのフルパス名(2004.3)。

INTEGRATE LOG FILE (pathName)

論理ミラーのセットアップ用コマンド。(詳細については 2004.3 追加/修正情報を参照してください。)

新しいバックアップ環境設定:

環境設定のバックアップページ(「バックアップ」テーマ)に「操作をキャンセル□回試行後」というオプションが追加されました(2004.3)。

ログファイルのサイズが 2GB に近づくと、自動的にバックアップが実行されるようになりました(2004.3)。

リストボックスで新しくなったこと

SCROLL LINES (*;object;position) *position* で指定した行がリストボックスの一番上に表示されるようにスクロールします(2004.1)。

追加された定数:

Get listbox information (*;object; Position listbox ver scrollbar)

Get listbox information (*;object; Position listbox hor scrollbar)

リストボックスのスクロールバー位置をポイント単位で返す定数が追加されました(2004.1)。

空の日付は 00/00/00 としてではなく、空白が表示されるようになりました(2004.1)。

スリーステートチェックボックスが表示できるようになりました(2004.1)。

列に関連付けられた配列の更新はすぐにリストボックスに反映されるようになりました (2004.1)。

4D Toolsで新しくなったこと

ストラクチャタブ: コンパイル済コードを削除するためのチェックボックスが追加されました(2004.1)。

追加されたオプション: **すべてのデータベースフォームを再生成**。変換データベースおよび修復を必要としているデータベースが対象です(2004.2)。(コピー&圧縮後に使用してください。)

4D Writeで新しくなったこと

WR ON EVENT および WR SET AREA PROPERTY コマンドに新しいパラメータが追加されました:

WR ON EVENT (area;wr on timer;method)

デフォルトの設定では 60 秒ごとにメソッドが実行されます。

間隔を変更するには: **WR SET AREA PROPERTY** (area;wr timer frequency;value)

4D Write ドキュメントから HTML ファイルへの変換が強化されました。(2004.2)

新しい印刷コマンド:

WR Print settings to BLOB

WR BLOB TO PRINT SETTINGS

プリントパラメータ、レイアウトの保存と設定をするためのコマンドが追加されました。(2004.2)

警告: 4D Write 2004.3 ではドキュメント形式が変わりました。以前のバージョンで開くことはできません。

ドキュメントの互換性

4D Write ドキュメントの内部フォーマットが変更されたため、4D Write 2004.3 (あるいは 4D Write 2003.7) ドキュメントを以前のバージョンで開くことはできなくなりました。

罫線の管理:

段落内罫線の管理が改良されました。(2004.3)

WR SET TEXT PROPERTY の定数 43 および 44 の名称が変更されています。

wr top border (43)新名称 **wr inside top border** (各段落の上に空白を設置)

wr bottom border (44)新名称 **wr inside bottom border** (各段落の下に空白を設置)

新しい定数:

wr top border (46) (段落全体の上に空白を設置)

wr bottom border (47) (段落全体の下に空白を設置)

ドキュメントを開く際にズームを適用:

新しい定数:

WR SET AREA PROPERTY (area; wr use saved zoom value; value)

値: 1: 保存されたズーム値を使用する; 0: 100%に復帰する。(2004.3)

タブの除去:

「タブ」ダイアログ画面の新しいボタン「全て削除」を使用することにより、選択されたテキスト範囲のタブをすべて取り除くことができるようになりました。(2004.3)

4D Viewで新しくなったこと

プロパティリストから表示される「詳細プロパティ」ダイアログボックスにより、表示、入力、選択/ドラッグ & ドロップ、トリガ、オプション、およびコードの自動生成ができるようになりました。(2004.1)

新しいコマンド:

PV Print settings to blob (area) -> blob (2004.1)

PV BLOB TO PRINT SETTINGS (area;blob) (2004.1)

4D Packで新しくなったこと

AP Add Table and Fields (tableName; fieldNamesArray; fieldTypesArray; fieldLengthsArray)

AP Get Templates (templateNameArray)

AP Create relation (sourceTableNum;sourceFieldNum;destTableNum; destFieldNum)

AP Create method (methodName; propertiesArray; methodCode {}; folderName {}) (以上、2004.1)

コンポーネント

Current method name 関数はコンポーネントメソッドでも動作するようになりました。

9 – リストフォーム Q&A

以下の内容は、4D S.A.テクニカルサポートスタッフおよび Yves Casquel によって執筆され、www.4d.frで Web 公開されているドキュメントから転用されたものです。

バージョン 2004 では、リストフォームの制御に顕著な改善が施されています。過去のバージョンとは異なり、リスト形式で表示される出力フォームの動作は基本的に標準化されました。

過去のバージョンには、3 種類のリストフォームがありました:

- ユーザモードのリストフォーム: リスト更新、複数行の選択ができる。
- **DISPLAY SELECTION** および **MODIFY SELECTION** コマンドで表示された出力フォーム: リスト更新ができない。
- サブフォームとして表示されたリストフォーム: 複数行の選択ができない。

それぞれのリストフォームは、動作が異なるため、限られた用途で使用されていました。

バージョン 2004 では、上記 3 種類のリストフォームは動作が統一されました。ただし、サブフォームとして表示されたリストフォームだけは多少の例外があります。

基本コンセプトの覚え書き

カレントセレクション:

データベースにアクセスする場合、レコードの集合を取り出してからデータを操作します。この集合がカレントセレクションです。カレントセレクションには、1 件、数件、あるいは全件のテーブルレコードが含まれることがあります。カレントセレクションは、それぞれのテーブルについて存在します。

カレントレコード:

一般的な例: 4D はカレントセレクションが確定すると、セレクションの先頭レコードをメモリに読み込みます。このレコードはカレントレコードと呼ばれます。

ある場合には、カレントレコードがない、またはカレントレコードがカレントセレクションに含まれない、ということもあり得ます。あるいは、カレントレコードが決まっても、メモリに読み込まれていないという場合があります。そのような場合、「カレントレコードポインタ」がそのレコードを指しています。

リストフォームのハイライトされた行:

表示されているリストフォームの選択モードプロパティ(後述)が「単一」の場合、ハイライトされたレコードはカレントレコードに対応します。

表示されているリストフォームの選択モードプロパティ(後述)が「複数」の場合、カレントレコードはユーザアクションもしくはプログラミング言語によるハイライトの変更には影響されません。

4D 2004のサブフォームプロパティおよび過去バージョンとの互換性

4D 2004 では、過去のバージョンにおけるサブフォームの動作を再現するために「空行をダブルクリック」というプロパティを設定することができます。



2004 に変換されたデータベースのプロパティ

2004 で作成されたデータベースのプロパティ

「空行をダブルクリック」オプションを「レコード追加」に設定することは、なるべく避けるようにしてください。そのようなインターフェースは、現行のシステム(OS)にそぐわないものです。

単一選択リストでカレントレコードがロードされないのはなぜですか。

リストフォームの含まれるウィンドウをシステム(OS)が再描画すれば、再描画のメカニズムによりカレントレコードがアンロードされることとなります。この場合、カレントレコードポインタはアンロードされたレコードを指しています。

この新しいメカニズムでは、アンロードされたカレントレコードの値を読み込むために **LOAD RECORD** コマンドを実行する必要があります。フォームによっては、On Load イベントでカレントレコードをロードする必要があるかもしれません。

なぜウィンドウの再描画でリストフォームのカレントレコードがアンロードされるのですか。

リストフォームの含まれるウィンドウをシステム(OS)が再描画すれば、再描画のメカニズムによりカレントレコードがアンロードされることとなります。この場合、カレントレコードポインタはアンロードされたレコードを指しています。

この新しいメカニズムでは、アンロードされたカレントレコードの値を読み込むために **LOAD RECORD** コマンドを実行する必要があります。

注記: ボタンオブジェクトがフォーカスを取得した場合も同じことが起きます。

表示されたリストフォームに最初からハイライトされた記録があるのはなぜですか。

リストフォームの選択モードによって動作が異なります:

- 単一

4D は自動的にカレントレコードをハイライトします。ただしカレントレコードはロードされていません。**GOTO SELECTED RECORD** ([table];0)コマンドを実行すると、カレントレコードがアンロードされるだけでなく、カレントレコードポインタもクリアされるので、ハイライトレコードはなくなります。詳細については、質問「リストフォームのハイライトをどうすればコマンドで制御できますか。」を参照してください。

典型的なケースでは前述のコマンドを **On Load** イベントで実行します。

- 複数:

はじめて表示されたリストフォームでレコードがハイライトされることはありません。カレントレコードはロードされておらず、On Clicked イベントで自動的にロードされることもありません。

注記:

リストフォームの選択モードは、次のいずれかの方法で設定します:

- **MODIFY SELECTION** および **DISPLAY SELECTION** コマンドの追加されたオプション。
- サブフォームのプロパティ設定。

リストフォームに空行が表示されなくなったのはなぜですか。

4D 2004 では、それぞれのシステム(OS)の推奨されたインタフェースに準拠するよう努力が払われ、リストフォームの動作が変更されました。

バージョン 2004 では、フォームの表示領域を埋めるのに満たない件数のレコードを表示している場合にのみ、リストフォームの最後に空行が表示されます。それ以上のレコード数がある場合、フォームの一番下までレコードが表示されます。

注記: バージョン 2003 では、常に余分な領域がリストフォームの最後に表示されていました。バージョン 2004 では、余分な領域ではなく、空行(レコードデータのない行)として処理されます。

リスト形式でレコードデータを表示するアプリケーションは、一般に空行のダブルクリックによるレコード追加を受け付けてはいません。

過去のバージョンの 4D では、リストフォームの空いた領域をダブルクリックすることによってレコードを追加することができました。バージョン 2004 で作成されたフォームの場合、この動作はサポートされません。

アップグレードされたフォームの場合、プロパティを変更しなければ、過去のバージョンにおける動作が再現されることとなります。

注記:

この場合、リストフォームに空行がなければ、レコードが追加できないこととなります。

空行ダブルクリックによるレコード追加を避けるべきなのはなぜですか。

この互換性プロパティは、過去のバージョンにおける動作を再現するためにバージョン2004に追加されているものです。バージョン2004で新しく作成したリストフォームでは利用することができません。

バージョン 2004 の新しいルック&フィールに合わせ、このオプションはなるべく使用しないでください。

このオプションを使用する場合、空行が表示されないリストフォームは、ダブルクリックでレコードが追加できなくなります。

空行のダブルクリック以外にレコードを追加する方法がありますか。

1) 次の標準アクションをボタンに割り当てることができます: サブレコード追加

この標準アクションは以前のバージョンから存在するものです。
バージョン2004では、動作の適用範囲が拡がりました: サブフォームに加え、**MODIFY SELECTION** コマンドで表示されたリストフォームにレコードを追加することもできるようになりました。

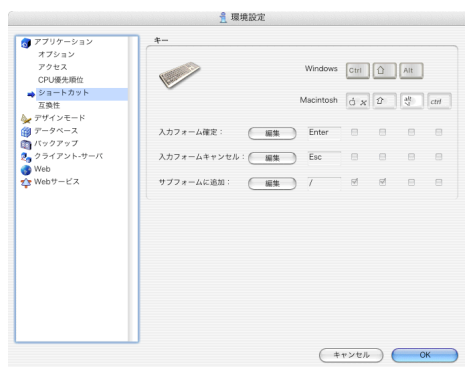
2) オブジェクトメソッドの定義されたボタンを配置することができます: メソッドでは、**ADD RECORD** コマンドを使用します。

注記: このようなボタンは、対象となるリストフォームによって配置する場所が異なります:

- サブフォームが組み込まれたフォームの場合:
メインフォームにサブフォームレコード追加ボタンが配置できます。
- サブフォームの場合(バージョン 2004 の新しい動作)
サブフォームのヘッダまたはブレイクエリアに追加ボタンが配置できます。
- リストフォームの場合(サブフォームではない):
リストフォームのヘッダ、ブレイクまたはフッタエリアに追加ボタンが配置できます。

3) プロジェクトメソッドの定義されたメニュー項目を作成することができます。 メソッドでは**CREATE RECORD** コマンドを使用します。 あるいは、サブフォームのセレクションにアクセスする適当なコマンドを使用します。

4) 環境設定のキーボードショートカットシステムを使用するという方法もあります:



注記:

コマンドでサブレコードの削除や更新をする場合:

DELETE SUBRECORD
MODIFY SUBRECORD
MODIFY RECORD
DELETE RECORD

上記の説明における対応する場所でそれぞれのコマンドを使用してください。

4D 2004に変換されたリストフォームの選択モードはどうなりますか。

次の表には、変換されたリストフォームに適用される「選択モード」プロパティの設定が示されています。

4D 2003 から 4D 2004 に変換されたデータベースの場合:

バージョン 2003 プロパティ	バージョン 2004 プロパティ
サブフォーム: データ入力可、ダブルクリック可	リスト更新可 行をダブルクリック: レコード更新 空行をダブルクリック: レコード追加 選択モード: なし
サブフォーム: 表示のみ	リスト更新不可 行をダブルクリック: 何もしない 空行をダブルクリック: 何もしない 選択モード: なし
サブフォーム: 選択可、ダブルクリック可	リスト更新不可 行をダブルクリック: レコード表示 空行をダブルクリック: 何もしない 選択モード: 単一
次のコマンドで表示されたリストフォーム: MODIFY SELECTION DISPLAY SELECTION	選択モード: 複数
ユーザモードのリストフォーム	選択モード: 複数

バージョン 2004 のリストフォームは、サブフォームを含め、レコードの選択モードが設定できるようになっています。提供されている選択モードは次のとおりです:

- 単一 (ひとつだけレコードが選択できる)
- 複数 (ひとつまたは複数のレコードが選択できる)
- なし (選択することはできない)

この新しいメカニズムが実装されたことにより、次のことができるようになりました:

ユーザは、矢印キー、キーボードショートカット、マウス操作などで単一または複数のレコードをハイライトすることができます。

選択モードが「なし」であれば、レコードをハイライトすることはできません。

デベロッパは、コマンドでリストフォームの選択モードを制御することができます。詳細については、質問「リストフォームのハイライトをどうすればコマンドで制御できますか。」を参照してください。

On Clicked イベントで自動的にカレントレコードがロードされなくなったのはなぜですか。

リストフォームの選択モードが下記いずれかに設定されている場合、質問のような現象が起こります:

- 複数
- なし

選択モード「複数」におけるクリック操作は、単純にハイライトを変更するだけのアクションです。選択モード「なし」でアクションを起こすことはできません。

過去のバージョンから変換されたデータベースは、リストフォームの選択モードによって動作が異なります。詳細については、質問「4D 2004 に変換されたリストフォームの選択モードはどうなりますか。」を参照してください。

On Displayed Detailが空行で発生するのはなぜですか。

On Display Detail フォームイベントは、リストフォームの行が表示される時には必ず発生するようになりました。したがってレコードが含まれない空行についてもイベントが発生することになります。空行には、他のプロセスやクライアントによって削除されたレコードの行も含まれます。こうした変更を踏まえ、リストフォームの表示が崩れないようにするためには、適切なコードを記述する必要があります。

Tip: 表示が崩れないようにするために次のような仕方でコードを記述することが推奨されています:

Case of

```

¥ (Form event= On Display Detail)
  If (Is record loaded (Current form table ->))
    レコードを表示している行の処理
  Else
    レコードを表示していない行の処理
  End if

```

End case

バージョン 2004 では、システム(OS)のガイドラインに準拠し、リスト下部のレコードがない部分についても、空行が描画されるようになりました。

サブフォームを選択しても点滅アイコンが表示されないのはなぜですか。

バージョン 2004 では、システム(OS)のガイドラインに準拠し、フォーカスを取得したサブフォームのグラフィックが見直されました。過去のバージョンのようにサブフォームの左上に三角アイコンが表示されることはなくなりました。

バージョン 2004 では、OS の基準に合った仕方でフォーカスを取得したサブフォームが表現されるようになりました。以前のインタフェースを擬似的に再現したい場合、フォームの左上にピクチャを配置しておき、サブフォームを組み込まれたフォームの **On Getting Focus** および **On Losing Focus** フォームイベントでそのピクチャを制御することができます。

リストフォームオブジェクト(フィールド/変数)が表示されないのですが。

リストフォームの下記のエリアにオブジェクトを配置した場合:

- ヘッダエリア
- 詳細エリア
- ブレイクエリア

オブジェクトは完全にそのエリアに含まれていなければなりません。オブジェクトが詳細エリアのマーカを超えることは許されません。エリアをまたがるオブジェクトは最適された表示メカニズムに干渉し、アピランスを崩す恐れがあります。

データベースをアップグレードする場合、事前にリストフォームのマーカ位置を確認するのが賢明です。

リストフォームアピランスが過去のバージョンと違うのはなぜですか。

オブジェクトを表示するためのメカニズムは、バージョン 2004 で最適化され、これまでよりも洗練されたアピランスが表現できるようになりました。

背景画像のように、上端がフォームのヘッダエリアで始まって下端がブレークエリアで終わるようなオブジェクトは、オブジェクトの高さを引き延ばす格好でひとつだけ描画されるようになりました。

リスト更新が動作しないように思えるのはなぜですか。

バージョン 2004 では、システム(OS)の違いに対応するため、クリックの処理が標準化されました。

最初のクリックで行が選択されます。

選択された行をもう一度クリックするとリスト更新モードになります。この場合、一定の間隔を空けてクリックすることにより、ダブルクリックの操作と区別することができます。

クリックの間隔が短い場合、それはダブルクリックの操作になります。クリックの間隔がダブルクリックよりも長い場合、2度目のクリックはシングルクリックとみなされます。

フィールドや変数のリスト更新をどうすればコマンドで開始できますか。

プログラミング言語に **EDIT ITEM** という新しいコマンドが追加されました。

このコマンドを使用すると、リストフォームおよびサブフォームの場合、指定された行の入力順の設定における最初のフィールドがリスト更新モードになります。

EDIT ITEM ({*;} "Subform name"; item)

item: 複数選択リストフォームにおける行の位置(順番)

省略した場合、最後にクリックされたカレント行に適用されます。

最後にクリックされたカレント行がない場合、最初の行に適用されます。

このコマンドを実行すると **On Selection Change** フォームイベントが発生します。

このコマンドを実行すると **On Load Record** フォームイベントも発生します。

リストフォームおよびサブフォームは、次のコマンドでもリスト更新モードにすることができます:

SET ENTERABLE ([table] ;True)

SET ENTERABLE (*;"Subform Name"; True)

SET ENTERABLE はサブフォーム側のフォームメソッドで実行しなければなりません。

クリックまたはコマンドでリスト更新モードに突入した行の変数をどうすれば初期化できますか。

バージョン 2004 における **On Detail Display** フォームイベントのメカニズム変更(質問「On Displayed Detail が空行で発生するのはなぜですか。」を参照してください。)に関連して新しいフォームイベントが設けられました。

On Load Record フォームイベントの基本メカニズム:

リストフォームでリスト更新モードが発動するときに発生するイベント。

このイベントを発生させる操作:

- 入力できるエリアを 2 度クリックするユーザアクションで更新モードに移行
- **EDIT ITEM** コマンドを使用して更新モードに移行

On Load Record フォームイベントを処理するためのプログラミング:

カレントレコードに関連のある変数がリスト更新モードで正しい値になるように、プログラムを記述することができます。実際、**On Load Record** フォームイベントでは、表示されようとしているカレントレコードに関連のあるすべての変数が更新されます。表示されている変数の値は、編集されようとしているカレントレコードの値との同期が外れている場合があるからです。

変数の配置されたリストフォームでレコードを更新する場合、このイベントの処理は事実上必須です。

リストフォームのハイライトをどうすればコマンドで制御できますか。

プログラミング言語でハイライトを制御する方法は、フォームプロパティで設定された選択モードによって異なります。

サブフォームにおけるハイライトは UserSet によるメカニズム(レコード選択)と干渉しないよう別個で管理する必要があります。したがってサブフォームのレコードをハイライトした場合、UserSet は作成されません。

選択モードによってプログラミングの方法が異なります。

-選択モード「単一」

GOTO SELECTED RECORD コマンドなどのカレントレコードを変更するコマンド、あるいは自動リレートが適用されるコマンドを利用してカレントレコードを変更します。結果として新しいカレントレコードがハイライトされます。

バージョン 2004.1 では、コマンドにゼロが渡せるようになりました:

GOTO SELECTED RECORD (table;0) (受け入れられる値: ゼロ)

ゼロを渡した場合、カレントレコードはアンロードされ、カレントレコードポインタがクリアされます。結果としてレコードのハイライトはリセットされます。

-選択モード「複数」

HIGHLIGHT RECORDS コマンドでレコードをハイライトすることができます。コマンドを実行すると指定したレコードがハイライトされ、カレントレコードポインタは変わりませんが、カレントレコードがアンロードされます。

注記:

リストフォームの選択モードは、次のいずれかの方法で設定します:

- **MODIFY SELECTION** および **DISPLAY SELECTION** コマンドの追加されたオプション。
- サブフォームのプロパティ設定。

ユーザが単一/複数のレコードをハイライトしたことをコマンドで調べるにはどうすればよいですか。

レコードがハイライトされたことは **On Selection Change** フォームイベントで検出することができます。ユーザがレコードをハイライトすると発生するイベントなので「On New UserSet」と呼べるかもしれません。

ハイライトされたレコードを調べる方法は、リストフォームの選択モードによって異なります:

- 選択モード「単一」

Selected record number 関数でセクションにおけるレコードの位置(順番)を調べることができます。

- 選択モード「複数」

GET HIGHLIGHTED RECORDS(2004)コマンドでリストフォームのハイライトされたレコードを調べることができます。このコマンドは選択モードが「単一」のリストフォームでは使用できません。

- 選択モード「なし」

ユーザはレコードをハイライトすることができません。

「On Selection Change」フォームイベントも発生しません。

注記:

リストフォームの選択モードは、次のいずれかの方法で設定できます:

- **MODIFY SELECTION**および**DISPLAY SELECTION**コマンドの追加されたオプション。
- サブフォームのプロパティ。

リストフォームで利用できる新しいフォームイベントは何ですか。

過去のバージョンでは、サブフォームのレコードを管理する場合、一般に「On Clicked」および「On Double Clicked」フォームイベントが用いられていました。

バージョン 2004 では、新しいフォームイベントがふたつ追加されました:

On Selection Change: リストフォーム、サブフォーム、およびリストボックスにおける選択行/カレントレコードもしくは選択状態/カレントセクションが変更されたとき。

詳細については次の質問を参照してください:

ユーザが単一/複数のレコードをハイライトしたことをコマンドで調べるにはどうすればよいですか。

On Load Record: リスト更新モードにおける編集モードに移行するとき。

詳細については次の質問を参照してください:

- クリックまたはコマンドでリスト更新モードに突入した行の変数をどうすれば初期化できますか。

- ユーザが単一/複数のレコードをハイライトしたことをコマンドで調べるにはどうすればよいですか。

ファイルプロシージャが実行されないときがあるのはなぜですか。

フォームがサブフォームとして使用されている場合、テーブルのファイルプロシージャは実行されません。そのような場合、環境設定の「互換性」ページにある「V3.x.x ファイルプロシージャを使用する」オプションが有効にされていたとしても、ファイルプロシージャは実行されません。

MODIFY SELECTION および **DISPLAY SELECTION** コマンドで表示されたフォームの場合、前述のオプションが有効にされていれば、ファイルプロシージャは実行されます。

注記:

バージョン 6.0 以降、ファイルプロシージャは過去のメカニズムとなっています。現在ではトリガを使用することが強く推奨されています。