



4th Dimension バージョン 2004.2 へようこそ。本ドキュメントは、今回のバージョンの新機能と変更点を以下のテーマ別に分けて説明します。

- 「環境設定」の管理：4D「環境設定」ファイルの場所を設定するメカニズムが変更され、データ管理に関連する新たな「環境設定」のオプションが2つ追加されました。
- 「デザイン」モード：「リストボックス」オブジェクト、「フォーム」および「メニュー」エディタについての管理方法が変更されました。
- Web サーバ：4D HTML タグから呼び出されるすべてのメソッド（4DVAR、4DLOOP その他を含む）は”4D ACTION、4D METHOD および 4DSCRIPT で利用可能”の属性を要求されるようになりました。
- ランゲージ：4th Dimension 2004.2 には、XML DOM ナビゲーションに関する新しいコマンドと新しいフォームイベント「On After Edit」が追加されました。また、既存の **PRINTERS LIST**、**PLATFORM PROPERTIES**、**Drop position**、**SET TABLE TITLES** および **SET FIELD TITLES** コマンドの機能もさらに拡張されました。
- 4D Tools：新しいオプションを利用してフォームを再生成することができるようになりました。
- 4D Write：HTML4 のサポートを強化するため、4D Write プラグインの HTML の書き出し機能には、CSS スタイルシートが採用されました。また、印刷設定を保存するため 2 つの新しいコマンドが使えるようになりました（**WR BLOB TO PRINT SETTINGS** および **WR Print settings to BLOB**）。

「環境設定」の管理

4D 2004.2 のアプリケーションの「環境設定」については、次の2点に変更されました。

- 「環境設定」ファイルの場所の変更
- 新しい2つの「環境設定」オプションの追加

「環境設定」ファイルの場所

2004.2のバージョンでは、4Dアプリケーション（4th Dimension、4D Server、4D Client、4D Runtime など）の「環境設定」ファイルの標準設置場所が変更されました。

以前のバージョンでは、これらのファイルがユーザ設定フォルダ内の「すべてのユーザ（All Users）」フォルダに納められていました（Windowsでは「Documents and Settings¥All Users¥Application Data¥」そしてMac OSでは、「Library: Application Support」）。ただし、Windows上の4D Clientでは、例外的にファイルが「Documents and Settings¥ユーザ名¥Application Data¥」に納められていました。

4th Dimension 2004.2では、標準として以下のガイドラインに沿ってファイルの場所を設定します。

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

この新しい動作により、マルチセッションシステム上での利用がより簡単になります。また、「すべてのユーザ（All Users）」フォルダがロックされているシステム上での4Dアプリケーションの使用を可能にします。

互換性

既存の「環境設定」フォルダとの互換性を保つため、4Dアプリケーションの起動時には次の手順が適用されます¹。

- アプリケーションは、まず「カレントユーザ」フォルダ（すべてのアプリケーション）の中に有効な「環境設定」フォルダがあるかどうかを探します。フォルダが既に存在する場合、このフォルダを使用します。

注：「環境設定」フォルダが有効かどうかは、「環境設定」フォルダ（“4D”）の中に“Lisences”フォルダがあるかどうかに基づいて判断されます。

1.この方法は、4D Clientを除くすべてのアプリケーションに適用されます。以後、環境設定フォルダは常にカレントユーザフォルダから検索されます。

■ 上記以外の場合、「All Users」フォルダ内を探します。

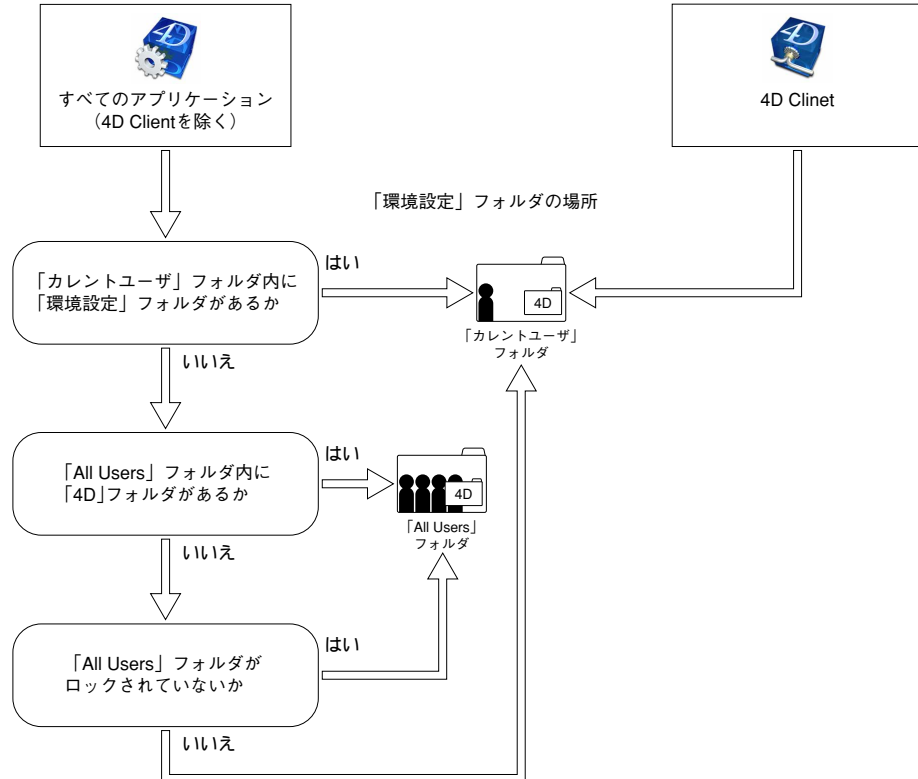
■ 有効な「環境設定」フォルダが見つかった場合、アプリケーションはこのフォルダを使います。

■ 既存のフォルダが見つからず、「すべてのユーザ」フォルダがロックされていない場合、この場所に「環境設定」フォルダが作成されます。

■ 「すべてのユーザ」フォルダがロックされている場合は、「カレントユーザ」フォルダに「環境設定」フォルダ作成されます。

このメカニズムにおいては、ユーザが手動で「環境設定」フォルダを移動し、設定場所を選択することも可能です。

以下の図は、4Dの環境設定フォルダの作成またはロードの流れを示しています。



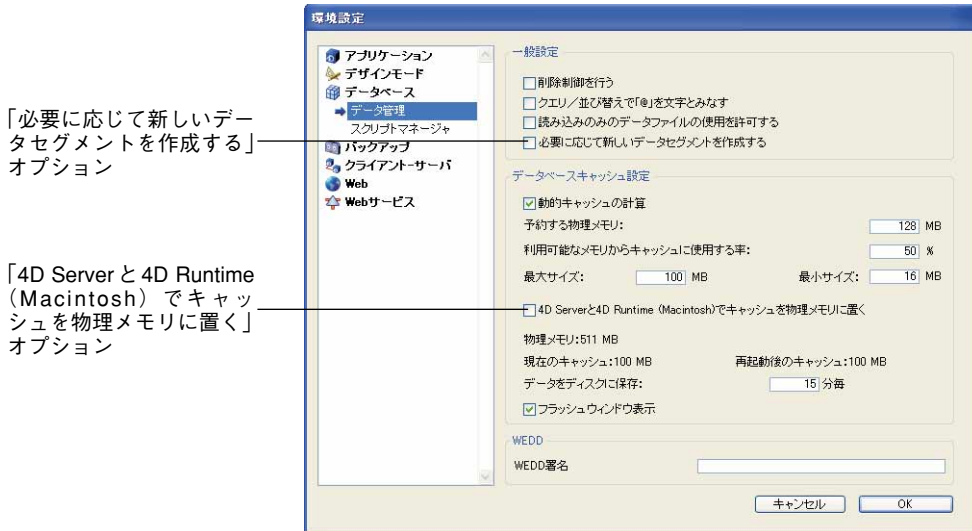
互換性の注意

バージョン2004.2から、(C:\¥Windows など) システムフォルダ内の4Dの環境設定フォルダは使用されなくなりました。この場所は4th Dimensionバージョン6.xとの互換のためにのみ利用されます。

アプリケーションがこの場所を利用している場合、その内容を手動で標準の場所に移動しなければなりません。

新しい「環境設定」

4th Dimension 2004.2では、[データベース/データ管理]ページに新たな2つのオプションが加わりました。「必要に応じて新しいデータセグメントを作成する」と「4D Serverと4D Runtime (Macintosh) でキャッシュを物理メモリに置く」です。



必要に応じて新しいデータセグメントを作成する

このオプションが選ばれた場合、データベースの作業中にデータファイル、もしくはそのセグメントのサイズが最大に達すると自動的に新しいデータセグメントが作成されます。ダイアログボックスは表示されません。データファイルおよび各セグメントの最大容量のデフォルト値は2GBで、上限は128GBです。

自動的に設定される各セグメントの最大容量は2GBで、これは最後に作成されたセグメント、もしくはそのデータファイルと同じ場所に保存されます。セグメント名は、データファイル名に基づいて規則的に設定されます。データファイル名がMyData.4DDの場合、最初のセグメントはMyData.4DS、2つ目のセグメントはMyData2.4DSようになります。

セグメントは手動でも作成できます。自動作成されたセグメントは、「ユーザ」モードの「セグメント」ダイアログボックスに表示されます。

保存先のディスクの容量不足により、自動的にセグメントを作成できない場合、警告ダイアログボックスが表示されます。この場合、新しいセグメントは手動で別の場所に作成しなければなりません。

4D Server と 4D Runtime (Macintosh) でキャッシュを物理メモリに置く

Mac 上では、4D Runtime (すべてのバージョン) や 4D Server によるデータベースの実行中、このオプションを使い、"強制的に"キャッシュを物理メモリ (RAM) に置くことができます。

Mac OS ではデフォルトで、必要な時にキャッシュメモリの一部もしくは全体が、マシンの仮想メモリに移されてしまう場合があります (仮想メモリとはハードディスクの中に特別に設けられたメモリの領域)。このメカニズムにより、4D Server や 4D Runtime で配付されたアプリケーションのパフォーマンスが著しく変更される恐れがあるため、このオプションを選択し、キャッシュを恒久的に物理メモリに保持しておくことができます。もちろん、マシンの設定 (RAM のサイズ、同時に起動しているアプリケーションの数、その他) は、この方針に対応するものでなければなりません。

注: このオプションは、シングルユーザ用の 4th Dimension アプリケーションには影響しません。

「デザイン」モード

4th Dimension 2004.2 の「デザイン」モードでは、「フォーム」、「リストボックス」、および「メニュー」エディタについての変更がありました。

フォーム

フォームのテスト

メソッドエディタからメソッドをテストできるように、フォームの実行も「デザイン」モードのフォームエディタから直接テストできるようになりました。

「フォーム」エディタからのテストを行なうには、「フォーム」エディタウィンドウ左上の「フォームを実行」ボタンをクリックします。

「フォームを実行」
ボタン



注：このボタンは、印刷フォームでは利用できません（印刷用詳細フォームと印刷用リストフォーム）。

このボタンをクリックすると、4th Dimensionは自動的に「ユーザ」モードに切り替わり、そのコンテキストでフォームを表示します（リストフォームではレコードリスト、詳細フォームではカレントレコードのページを表示）。フォームはメインプロセスで実行されます。

注：ユーザロックアイコンはウインドウの右上に移動しました。

リサイズの制限

2003.xもしくはそれ以前のバージョンから2004.2へ更新されたフォームの「プロパティリスト」（「ウインドウサイズ」テーマ）において、「制約付き」という新しい互換性のオプションが選択できるようになりました。

このオプションは、前に使用された“サイズ変更可”フォームプロパティの動作をするために使われます。このオプションを選択しないと、オブジェクトのリサイズやウインドウサイズの制限を行なうメカニズムが機能しません。この場合、ユーザは自由にフォームウインドウのサイズを変更することができます。ただし、ウインドウ内のオブジェクトのリサイズや移動は行われません。最小/最大、もしくは固定されたサイズのプロパティやオブジェクトのリサイズプロパティは無視されます。

これは、過去のバージョンの4th Dimensionの動作に対応するもので、特定のインターフェースにおける互換性のためにのみ使用されるものです。4D 2004バージョンで作成されたデータベースの動作は選択されたオプションに対応します（標準モード）。

リストボックスのドラッグ&ドロップ操作

1つのリストボックスから別のリストボックスや4Dオブジェクトへ（もしくはその逆方向へ）行をドラッグ&ドロップすることが可能になりました。

この操作には、以下の原則が適用されます。

■ リストボックスの行のみについての操作が可能であり、カラムについての操作は無効です（ただし、1つのリストボックス内でのカラム移動は可能）。

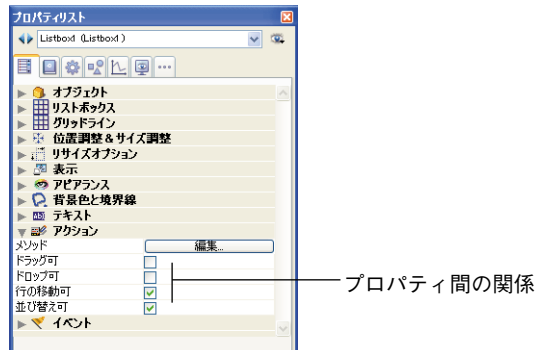
- ドラッグ&ドロップ操作は、1行ごとのみの移動に有効で、まとめて選択された行については現在のところ、対応していません。
- リストボックスのドラッグ&ドロップ操作は、4th Dimension 標準のドラッグ&ドロップメカニズムによって管理されます。
 - 「リストボックス」オブジェクトの「ドラッグ可」および「ドロップ可」プロパティ
 - 「On Drop」および「On Drag Over」フォームイベント
 - **DRAG AND DROP PROPERTIES** コマンドおよび **Drop position** 関数。**Drop position** 関数は、ドロップされたカラム数を検索するための新たなパラメータも受け入れるようになりました。この件についての詳細は、後述の「Drop position」の節を参照して下さい。

注：ドラッグ&ドロップ操作のプログラムは、「リストボックス」のオブジェクトメソッドで実行することができます。

4th Dimension のドラッグ&ドロップ操作の詳細については、『4th Dimension ランゲージリファレンスマニュアル』の「ドラッグ&ドロップコマンド」の章を参照してください。

- 「ドラッグ可」、「ドロップ可」、「行の移動可」プロパティ間の関係

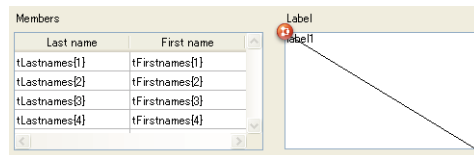
リストボックスでは、これら3つのプロパティを選択することができます。



これらプロパティの組み合わせによる結果は以下のようになります。

	組み合わせ1	組み合わせ2	組み合わせ3
ドラッグ可	-	○	○
ドロップ可	-	○または-	○または-
行の移動可	○	○	-
備考	リストボックスの操作は過去のバージョンの4th Dimensionと同一。互換性上の理由からこれをデフォルト設定とする。	行の移動では、「行の移動可」オプションが優先される。このため、ドラッグ操作はできない。「ドロップ可」オプションを選択すると、オブジェクトをリストボックス上にドロップすることができる。	ドラッグ&ドロップ操作が有効な状態（アクティベートされている状態）。「ドロップ可」オプションを選択するとリストボックス内でのドラッグ&ドロップ操作も同時に可能になる。

- ▼ リストボックスからドラッグしてきたデータでテキストエリア（例えばlabel1）を埋める場合。



リストボックスの中身は、[Members]テーブルから生成されます。

- テキストエリアの「On Drag Over」および「On Drop」イベントを有効にする。
- リストボックスの「ドラッグ可」プロパティを有効にする。
- テキストエリアの「ドロップ可」プロパティを有効にする。

label1のオブジェクトメソッドは次の通りです。

Case of

¥(Form event=On Drag Over)

DRAG AND DROP PROPERTIES(\$source;\$arrayrow;\$processnum)

If(\$source=Get pointer("listbox1"))

\$0:=0`ドロップが受け入れられた

Else

\$0:=-1

End if

¥(Form event=On Drop)

DRAG AND DROP PROPERTIES(\$source;\$arrayrow;\$processnum)

QUERY([Members];[Members]LastName=tLastnames{\$arrayrow})

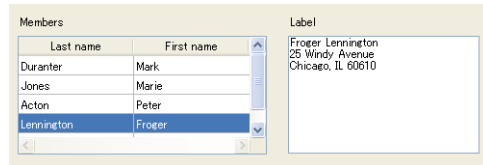
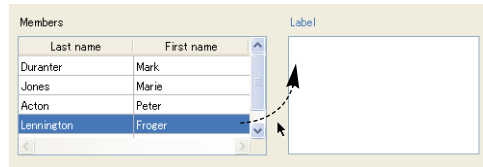
If(Records in selection([Members])#0)

label1:=[Members]FirstName+" "+[Members]LastName+
Char(Carriage return)+[Members]Address+
Char(Carriage return)+[Members]City+" "+
 +[Members]State+" "+[Members]ZipCode

End if

End case

これで以下のアクションを実行することができるようになりました。

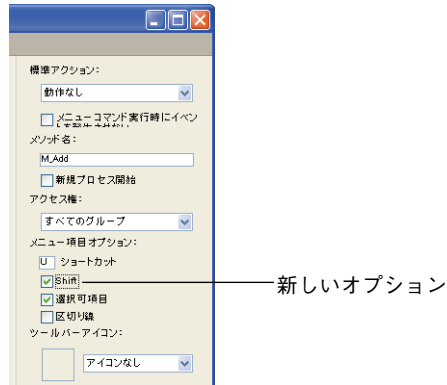


▼ リストボックスへのドロップ操作についての例は、**Drop position** 関数の記述を参照してください。

注：2004.2の4th Dimensionでは、リストボックスのセルを編集モードへ変更する方法が改訂されました。間隔をおいて2回クリックする方法は、操作がより簡単になりました。

Shift キーをショートカットに割り当てる

カスタムメニューコマンドに割り当てられたショートカットにShiftキーを追加することができるようになりました。方法としては、メニューエディタのオプションで「Shift」を選択するだけです。



この新機能により、作成したメニューバーで利用できるショートカットの数が2倍になります。ショートカットには常にCtrlキー（Windowsの場合）もしくはCommandキー（Mac OSの場合）が割り当てられる点に留意してください。「Shift」オプションを選択すると、Windowsで定義されるショートカットは、Ctrlキー＋Shiftキー＋文字、Mac OSではCommandキー＋Shiftキー＋文字となります。

注：Shiftキーをショートカットに割り当てる場合、A-Zの文字を使用すべきです。それ以外の文字は、キーボード配列とShiftキーの組み合わせで、指定した文字が使用されない場合があります。

ローマ字以外の文字のサポート

4th Dimension 2004.2では、ローマ字以外の文字のサポート範囲が拡張されました。

4th Dimension のアプリケーションでは、以下のスクリプトを利用できます。

Arabic、Simplified Chinese、Traditional Chinese、Croatian、Cyrillic、Eastern Europe、Greek、Hebrew、Icelandic、Japanese、Korean、Romanian、Thai、US-European、そしてVietnameseです。これらのスクリプトで、約60の言語にアクセスすることができます。

4th Dimension は、データ入力、表示および処理（例えばソートなど）で使用するシステム言語に対応するスクリプトを自動的に選んで使用します。

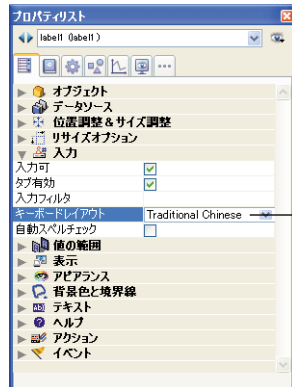
注：これら各言語に対する4th Dimensionの互換性については、4Dの検証結果に基づいて発行された『Language Certification Matrices (<http://www.4d.com>)』を参照してください。

「キーボードレイアウト」プロパティ

フォームで入力可能な以下のオブジェクトについては、特定の「キーボードレイアウト」プロパティを指定できるようになりました。

- フィールド
- 入力可能な変数
- 階層リスト
- リストボックス
- コンボボックス

プロパティリストの中に配置されたこのプロパティにより、システムと異なる言語スクリプトを使用してデータ入力および表示を行なうことができます。



「キーボードレイアウト」プロパティ

このプロパティを使用すると、入力された文字を内部変換する際“強制的に”指定したスクリプトが使用されます。またこのプロパティにより、オブジェクトにカーソルが入ったとき、キーボードが自動的に変更されます。

このプロパティを設定しないで、入力に使用するカレントキーボードをユーザが変更した場合、フォームに入力することはできても、入力された文字は正しく保存されません（カーソルがエリアを離れると文字が消える）。

重要：このプロパティは、データの入力および表示のみに有効です。データ処理（ソート、クエリなど）には、常にシステムのカレントスクリプトが使われます。

Web サーバ

"4DACTION...で利用可能"属性の拡張

HTTP リクエストによりコールされるメソッドを実行するために、"4DACTION、4DMETHOD および 4DSCRIPT で利用可能" プロパティのチェックを必要とするセキュリティメカニズムが拡張されました。

以前のバージョンの 4th Dimension では、4DACTION、4DMTEHOD および 4DSCRIPT でコールされるメソッドのみが、このルールの適用を受けていました。

4th Dimension 2004.2 からこのオプションは 4DVAR や 4DHTMLVAR からコールされるメソッドにおいてもチェックする必要があります。



注：これらのタグにより実行されるメソッドからコールされるサブメソッドには、この設定を明示的に行う必要はありません。

ランゲージ

4th Dimension のランゲージにおいて、次の点が変更されました。

- **Drop positon** 関数がパラメータを受け入れるようになりました。
- **PLATFORM PROPERTIES** コマンドが、カレントのシステム言語を返すようになりました。
- Mac OS 上で、**PRINTER LIST** コマンドが追加情報を返すようになりました。
- 「XML」テーマに3つの新しいDOMナビゲーションコマンドが追加されました。
- 新たな「On After Edit」フォームイベントを使い、カットやペーストのような編集イベントなど、入力エリアにおけるすべての変更が管理できるようになりました。

■ **SET DATABASE PARAMETER** および **GET DATABASE PARAMETER** コマンドが新しいセレクタを受け入れるようになりました。

Drop Position

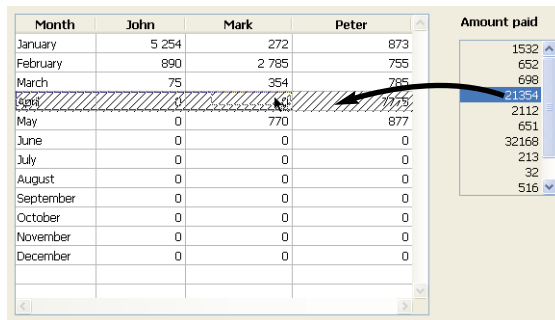
Drop Position{(columnNum)} → 数値

引数	タイプ	説明
columnNum	倍長整数	← カラム番号。または最後のカラムを越えた場合は-1
戻り値	数値	← ドロップ先の要素番号（配列・リストボックス）もしくは、項目位置（リスト）。ドロップ位置が最後の配列要素もしくはリスト項目を越えた場合は-1。

テーマ：ドラッグ&ドロップコマンド

Drop position 関数は、ドロップされたカラムの番号を返す引数<columnNum>を受け入れるようになりました。このコマンドの目的は、「リストボックス」オブジェクトで実行されたドロップ操作の管理です。このコマンドについての詳細は、後述の「リストボックス内のドラッグ&ドロップ操作」の節を参照してください。

▼ 次の例題では、支払い金額を月と支払った人ごとに分類しなければなりません。これは、スクロール可能なエリアからのドラッグ&ドロップ操作で行ないます。



リストボックスのオブジェクトメソッドには、次のコードが含まれています。

Case of

¥(Form event=On Drag Over)

DRAG AND DROP PROPERTIES(\$source,\$arrayrow,\$processnum)

If(\$source=Get pointer("SA1"))`スクロールエリアからドロップされたら
\$0:=0

Else

```

                                $0:=-1 `ドロップを拒否する
                                End if
                                ¥(Form event=On Drop)
                                DRAG AND DROP PROPERTIES($source;$arrayrow;$processnum)
                                $rownum:=Drop position($colnum)
                                If($colnum=1)
                                    BEEP
                                Else
                                    Case of `ドロップされた値を追加する
                                        ¥($colnum=2)
                                            John{$rownum}:=
                                                John{$rownum}+SA1{$arrayrow}
                                        ¥($colnum=3)
                                            Mark{$rownum}:=
                                                Mark{$rownum}+SA1{$arrayrow}
                                        ¥($colnum=4)
                                            Peter{$rownum}:=
                                                Peter{$rownum}+SA1{$arrayrow}
                                    End case
                                DELETE ELEMENT(SA1;$arrayrow) `エリアのアップデート
                                End if
                                End case

```

PLATFORM PROPERTIES

PLATFORM PROPERTIES (platform{; system{; machine{; language{}}})

引数	タイプ	説明
platform	数値	← 1= 68K ベースの Macintosh (旧式)、 2=Power Macintosh、3= Windows
system	数値	← 起動中のバージョンによる
machine	数値	← 起動中のバージョンによる
language	数値	← 使用中のシステムによる

テーマ：システム環境

新しい任意の引数 <language> は、起動しているデータベースのシステムのカレントランゲージを検出するために使用します。

コード	ランゲージ
1	Arabic
2	Bulgarian
3	Catalan
4	Chinese
5	Czech
6	Danish
7	German
8	Greek
9	English
10	Spanish
11	Finnish
12	French
13	Hebrew
14	Hungarian
15	Icelandic
16	Italian
17	Japanese
18	Korean
19	Dutch
20	Norwegian
21	Polish
22	Portuguese
24	Romanian
25	Russian
26	Croatian
26	Serbian
27	Slovak
28	Albanian
29	Swedish
30	Thai
31	Turkish
33	Indonesian
34	Ukranian
35	Belarusian
36	Slovenian
37	Estonian
38	Latvian
39	Lithuanian
41	Farsi

コード	ランゲージ
42	Vietnamese
45	Basque
54	Afrikaans
56	Faeroese

注：コマンドがシステム言語を認識できなかった場合、9の値（English）が返されます。

PRINTERS LIST

PRINTERS LIST (namesArray.{;altNamesArray{;modelsArray}})

Mac OS 上では、このコマンドの2番目のパラメータが、ユーザ編集可能な各プリンタのカスタム名を返します。このカスタム名は、例えばダイアログボックスの中などで使用することができます。

新しいDOM コマンド

XMLストラクチャ内におけるDOMナビゲーションを完成させるために、3つの新規コマンドが採用され、「XMLコマンド」テーマが強化されました。

既存「XML」テーマのDOMコマンド名が変更された点に留意してください（後述の「DOMコマンド名の変更」参照）。

DOM Get parent XML element

DOM Get parent XML element (elementRef{; parentElemName

{; parentElemValue}}) →文字列

引数	タイプ		説明
elementRef	文字列	→	XML要素の参照
parentElemName	文字列	←	親XML要素の名称
parentElemValue	文字列	←	親XML要素の値
戻り値	文字列	←	親XML要素の参照（16文字）

DOM Get parent XML element 関数は、<elementRef>に渡されたXML要素参照に対応する要素の、親要素のXML要素参照を返します。このXML要素参照は他のXML解析関数で使用します。

オプションの引数<parentElemName>および<parentElemValue>を渡すと、それぞれ、該当する親要素名および値を受け取ります。

コマンドが正常に実行されると、システム変数OKには1が代入されます。そうでない場合は、0が代入されます。

DOM Get Previous Sibling XML element

DOM Get Previous Sibling XML element(elementRef{; siblingElemName {; siblingElemValue})→文字列

引数	タイプ	説明
elementRef	文字列	→ XML要素の参照
siblingElemName	文字列	← 兄弟XML要素の名前
siblingElemValue	文字列	← 兄弟XML要素の値
戻り値	文字列	← 兄弟XML要素の参照 (16文字)

DOM Get Previous Sibling XML element関数は、<elementRef>に渡されたXML要素参照に対応する要素の、前の兄弟要素のXML要素参照を返します。このXML要素参照は他のXML解析関数で使用します。

オプションの引数<siblingElemName>および<siblingElemValue>を渡すと、それぞれ該当する兄弟要素名と値を受け取ります。

このコマンドは、子要素間のナビゲーションにも用いられます。

兄弟XML要素が最初の兄弟要素より前にある場合、システム変数OKには0が代入されます。

コマンドが正常に実行され、<elementRef>で参照される要素がストラクチャの最初の「子」ではない場合、システム変数OKには1が代入されます。エラーが発生した場合や<elementRef>で参照される要素がストラクチャの最初の「子」である場合、システム変数OKには0が代入されます。

DOM Get Last Child XML element

DOM Get Last Child XML element(elementRef{; childElemName {; childElemValue})→文字列

引数	タイプ	説明
elementRef	文字列	→ XML要素の参照
childElemName	文字列	← 子XML要素の名前
childElemValue	文字列	← 子XML要素の値
戻り値	文字列	← XML要素の参照 (16文字)

DOM Get Last Child XML element関数は、<elementRef>に渡されたXML要素参照に対応する要素の、最後の「子」を返します。このXML要素参照は他のXML解析関数で使用します。

オプションの引数<childElemName>および<childElemValue>を渡すと、それぞれ、該当する「子」要素名と値を受け取ります。

コマンドが正常に実行されると、システム変数OKには1が代入されます。そうでない場合は0が代入されます。

▼ 親ルートの最後のXML要素のリファレンスの取得

```
C_BLOB(myBlobVar)
C_STRING(16;$ref_XML_Parent;$ref_XML_Child)
C_TEXT($childName;$childValue)
DOCUMENT TO BLOB("C:¥¥import.xml";myBlobVar)
$ref_XML_Parent:=DOM Parse XML variable(myBlobVar)
$ref_XML_Child:=DOM Get Last Child XML element
($ref_XML_Parent;$childName;$childValue)
```

DOM コマンド名の変更

4D 2004.2では、名称を統一して明確にするため、既存の2つのXMLコマンド名を変更しました。

以前の名称	4D 2004.2で改訂された新規名称
DOM Get First XML element	DOM Get First Child XML element
DOM Get Next XML element	DOM Get Next Sibling XML element

新しい「On After Edit」フォームイベント

4th Dimension 2004.2では、「On After Edit」という新たなフォームイベントが利用できるようになりました。このイベントを有効にすると、入力可能エリア（フィールドもしくは変数）の内容が変更されるたびに、いかなる操作による変更であっても、このイベントが生成されます。あらゆるタイプの“入力可能”データが対象になります（文字列、テキスト、日付、時間、数値、ピクチャなど）。

このイベントは、次のアクションにより発生します。

■ 「ペースト」、「カット」、「削除」、「取り消し」など、内容の修正を行なう標準編集アクション

■ 値のドロップ操作（ペーストアクション）

■ ユーザによるすべてのキーボード入力。この場合、「On Before Keystroke」および「On After Keystroke」イベントが使用される場合は、これらのイベントの後に「On After Edit」イベントが生成されます。

■ ユーザアクションをシミュレートするランゲージコマンドを使用して行なったすべての修正（例：POST KEY や POST EVENT）

以下のアクションでは、このイベントが生成されませんので注意してください。

■ 「コピー」や「すべてを選択」のようにエリアの内容を修正しない編集アクション

■ 値のドラッグ操作（コピーアクション）

■ プログラムにより行われたすべての内容編集。ただし、前述のコマンドを除く。

このイベントは、長過ぎるテキストのペースト作業を防止したり、特定文字のブロック、もしくはパスワードのフィールドがカットされないようにするなど、ユーザアクションのコントロールに使用できます。

注：ほとんどの場合において、制限の多い既存の「On After Keystroke」イベントよりも、この新規イベントの使用をお勧めします。

SET DATABASE PARAMETER、Get database parameter

SET DATABASE PARAMETER({table;}selector; value)

Get database parameter({table;}selector) → 倍長整数

<selector>に新しい定数が利用できるようになりました。

セレクタ	値	適用範囲
Real Display Precision	32	4D アプリケーション

■ セレクタ=32 (Real Display Precision)

説明：実数を扱う場合、有効数字以外の数は、実際の画面表示のアルゴリズムに基づき末尾何桁かが切り捨てられます。この値は、カレントアプリケーションおよびセッションに合わせて設定されます。

デフォルトでは、このオプションの値は4です。0の値は、デフォルト値が使用され、セッション中にパラメータが修正されなかったことを示します。

歴史的な理由から、4th Dimensionは実数を10バイトに格納しますが、処理中にこの数字8バイトに変換します。この処理は完全に透過的で計算にも影響しません。ただし、特定の計算結果が予期した通りには表示されない場合があります。例えば、4.1-4.09の計算で表示される結果は0.009999999999999780000ですが、0.01を検索する際には正しい値が検索されます。

4Dは次のような方法で実数を表示します。

例えば、計算から得られた値が8.97499999999996158だとします(期待する値は通常8.975)。アルゴリズムに基づき、デフォルトでは末尾4桁(6158)を切り捨てた後、残った最後の数字が0もしくは9でないかを確認します。0の場合、アルゴリズムにより先頭の0以下すべての端数が切り捨てられます。値が9の場合、先頭の9の前の位の数字を繰り上げます。例えば、この例題では、8.97499999999996158が8.975になります。

例えば8.974999999999986158のような結果が得られる場合もあります。この場合、末尾の数字(最後の4桁を切り捨てた後)が0でも9でもないため、値は正しく四捨五入されません。

あなたのデータベースの特徴に合わせて、特別な精度アルゴリズムを使用して桁数に調整して四捨五入が行なわれるようにしたい場合があります。この場合、データベース固有の<value>を渡すことができます。0(4Dの内部で選択される値)を除く数字は、精度アルゴリズムにより切り捨てられる桁数を表します。

この設定は数値の表示やその際の内部処理には影響しません。

注:以前のバージョンの4th Dimensionでは、4D Customizer Plusでこの設定を利用することができました。

SET FIELD TITLES、SET TABLE TITLES

SET FIELD TITLES(table | subtable; fieldTitles; fieldNumbers(*))

引数	タイプ	説明
テーブルまたはサブテーブル	テーブルまたはサブテーブル	→ フィールド名を設定するためのテーブルもしくはサブテーブル
フィールド名	文字列配列	→ 新規フィールド名
フィールド番号	数値配列	→ 実際のフィールド番号
*	*	→ *を渡した場合=フォーミュラエディタでパーチャルストラクチャを使用 省略した場合=使用しない

SET TABLE TITLES(tableTitles; tableNumbers{; *})

引数	タイプ	説明
tableTitles	文字列配列→	ダイアログに表示されるテーブル名
tableNumbers	数値配列	→ 実際のテーブル番号
*	*	→ *を渡した場合=フォームラエディタでバーチャルストラクチャを使用 省略した場合=使用しない

SET FIELD TITLES および **SET TABLE TITLES** コマンドでは、任意の引数<*>を受け付けるようになりました。

この引数を省略した場合、これらのコマンドによって設定されたカスタム名は、4Dで実行されるフォーミュラに適用されません（2004.2の新しいデフォルト動作で、2003.xと同一のもの）。

この引数を渡した場合、このカスタム名は、4th Dimensionで実行されるフォーミュラに適用されます（4th Dimension 2004.0および2004.1のデフォルト動作）。

この場合、4Dランゲージインタプリタが“禁止”している文字（例えば、-?!）をカスタム名に含めることはできません。

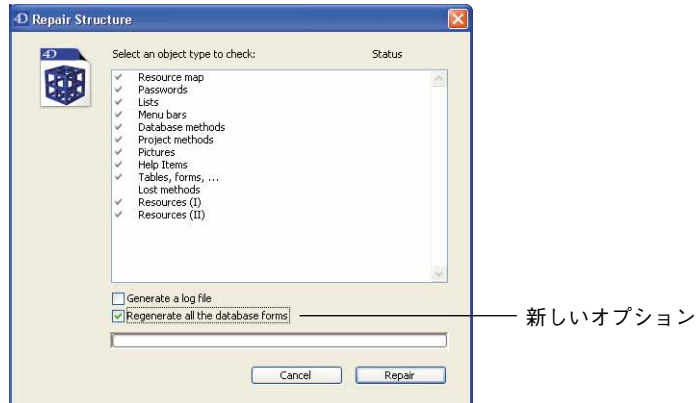
互換性についての注意：4D 2004のバージョンでは、SET FIELD TITLES および SET TABLE TITLES コマンドにより設定されたカスタムフィールド名とテーブル名（バーチャルストラクチャ）をフォーミュラエディタで使用するためのオプションが追加されました。この新機能ではカスタム名と4Dインタプリタとの互換性が要求されるため、カスタム名を定義するための規定もさらに厳しくなりました。

フォーミュラエディタでバーチャルストラクチャが使用されない制限を解くために開発者は、引数<*>を指定する必要があります。4th Dimension 2003.xと同様、デフォルトではフォーミュラエディタにバーチャルストラクチャは使用されません。

4D Tools : フォームの更新

新しい4th Dimensionのバージョンでデータベースを開いたとき、特に古いデータベースを開いた際、フォームによっては操作に異常がある場合があります。

この問題を解決するために、4D Toolsの「ストラクチャ修復」ウィンドウで「すべてのデータベースフォームを再生成」オプションを利用することができます。



注：このウインドウを表示するためには、[ストラクチャ]ページの「修復...」ボタンをクリックします。

このオプションにより、単独の操作でデータベースのすべてのフォームを更新および保存することができます。このオプションが選択されると、4th Dimensionは修復作業中のフォームを“強制的に”確認および保存します。

オリジナルのデータベースに損傷したフォームが含まれている場合、このオプションにより問題が生じる恐れがあります。したがって4D Toolsを使用する際には、事前にデータベースのコピー作成し、それを圧縮した後に修復することをお勧めします。

4D Write

2004.2バージョンの4D Writeには次のような新機能が含まれています。

- 向上された変換機能（4D Write ドキュメントからHTMLへの変換）
- 新たなプリント設定管理コマンド

HTML へのドキュメント変換

4D Write ドキュメントからHTMLに変換するための機能が向上しました。今回の4D Writeでは、スタイルおよびフォーマットの変換にCSSが採用されているため、変換されたドキュメントは、オリジナルに大変類似したものになります。

以下の表は、現在サポートされる要素を示します。

属性	HTML への変換
標準スタイル	Yes
太字	Yes
斜体	Yes
取り消し線	Yes
上付き	Yes
下付き	Yes
影付き	No
下線	Yes (一本線)。それ以外の下線は、すべて一本線に変換される
フォント	Yes
フォントサイズ	Yes
背面および前面の色	Yes (テキストと罫線)。取り消し、影付きテキストおよび下線は変換されない
行頭記号	Yes ("黒い四角" と "白い丸")。それ以外の場合は、標準タイプに置き換えられる
配置	Yes
間隔	Yes
インデント	Yes
罫線	Yes
タブ	Yes。スペースに置き換えられる (nbsp)
テキスト中のピクチャ	Yes (この変換には QuickTime のインストールが必要)
バックグラウンドピクチャ	No
ハイパーリンク	Yes (標準リンク)。ドキュメントを開いたり 4D メソッドを実行するリンクは変換されない

新しい印刷コマンド

新しい2つのコマンドにより、プログラムで 4D Write の印刷設定を管理できるようになりました。追加されたのは、**WR Print settings to BLOB** および **WR BLOB TO PRINT SETTINGS** (「WR Printing」テーマ) です。

WR Print settings to BLOB

WR Print settings to BLOB (area) → BLOB

引数	タイプ	説明
area	倍長整数	→ 4D Write のエリア
戻り値	BLOB	← 印刷設定を含む BLOB

WR Print settings to BLOB コマンドは、4D Write エリア <area> の現在の印刷設定を BLOB に格納します。<area> とは、プラグインウィンドウやフォーム上のエリア、オフスクリーンエリアを含みます。

BLOB には、印刷に必要な設定がすべて納められます。

■ レイアウトパラメータ（用紙、方向、倍率）

■ 印刷パラメータ（部数、用紙トレイなど）

注：Windows 上では、BLOB に格納される設定にプリンタが含まれます。

このコマンドは、プリンタの機種や利用可能な印刷設定を問わず、4D Write エリアの印刷設定を保存することができます。返される BLOB の内容はプログラムから変更しないでください。この BLOB は、**WR BLOB TO PRINT SETTINGS** コマンド（もしくは、4D Pack の **AP BLOB to print settings** コマンド）でのみ使用することができます。

例えば **WR SET PRINT OPTION** コマンドを使用して現在の印刷設定を一時的に変更する前に、**WR Print settings to BLOB** 関数を用いて現在の設定を保存しておくことができます。また、印刷が終了後に、**WR BLOB TO PRINT SETTINGS** コマンドを使用して現在のパラメータに戻すことができます。

BLOB が正常に生成された場合は、システム変数 OK に 1 を代入します。それ以外の場合は 0 を代入します。プリンタが選択されていない場合、エラーコード 1014 が返されます。

参照

WR BLOB TO PRINT SETTINGS

WR BLOB TO PRINT SETTINGS

WR BLOB TO PRINT SETTINGS(area; printSettings{; paramType})

引数	タイプ	説明
area	倍長整数	→ 4D Write のエリア
printSettings	BLOB →	印刷設定を含む BLOB
paramType	倍長整数	→ 0=レイアウトおよび印刷、1=印刷

WR BLOB TO PRINT SETTINGS コマンドは、4D Write エリア <area> の現在の印刷設定を、BLOB に格納された <printSettings> で置き換えます。

この <area> は、プラグインウィンドウやフォーム上のエリア、オフスクリーンエリアを含みます。ただし、4D Write の印刷設定を管理するメカニズムにより、0 を引数 <area> に渡すことによりすべてのエリアに適用することはできません。

この<printSettings>BLOBは、**WR Print settings to BLOB**関数により（もしくは、**4D Pack AP Print settings to BLOB**関数により）生成しなければなりません。

<printSettings>BLOBには、2種類のパラメータが含まれます。

- レイアウトパラメータ（用紙、方向、倍率）
- 印刷パラメータ（部数、用紙ソース等）

注：Windows上では、BLOBに格納される設定にプリンタが含まれます。

<paramType>に0を渡した場合、もしくはこの引数を省略した場合、すべてのパラメータが使用されます。1を渡した場合、印刷パラメータのみが適用されます。レイアウトパラメータは変更されません。

新しい印刷設定は<area>に表示されるドキュメントに適用されます。

注：WindowsとMac OSでは、印刷設定のフォーマットが異なります。このため、<printSettings>BLOBと2つのプラットフォーム間における互換性は保証されません。

BLOBが正常にロードされている場合、システム変数OKには1が代入され、そうでない場合は0が代入されます。

プリンタが選択されていない場合、エラーコード1014が生成されます。<printSettings>BLOBに有効な印刷設定が含まれていない場合は、エラーコード1074が生成されます。

参照

WR Print settings to BLOB

