

4th Dimension 2004

アップグレード
Windows[®] and Mac OS[®]



4th Dimension 2004 アップグレード

Windows® and Mac OS®

Copyright© 1995 - 2005 4D SA

All rights reserved.

このマニュアルに記載されている事項は、将来予告なしに変更されることがあり、いかなる変更に関しても 4D SA は一切の責任を負いかねます。このマニュアルで説明されるソフトウェアは、本製品に同梱の License Agreement（使用許諾契約書）のもとでのみ使用することができます。

ソフトウェアおよびマニュアルの一部または全部を、ライセンス保持者がこの契約条件を許諾した上での個人使用目的以外に、いかなる目的であれ、電子的、機械的、またどのような形であっても、無断で複製、配布することはできません。

4th Dimension、4D Server、4D、4D ロゴ、4D ロゴ、およびその他の 4D 製品の名称は、4D SA の商標または登録商標です。

Microsoft と Windows は Microsoft Corporation 社の登録商標です。

Apple, Macintosh, Mac, Power Macintosh, Laser Writer, Image Writer, ResEdit, QuickTime は Apple Computer Inc. の登録商標または商標です。

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

注意

このソフトウェアの使用に際し、本製品に同梱の License Agreement（使用許諾契約書）に同意する必要があります。ソフトウェアを使用する前に、License Agreement を注意深くお読みください。

第 1 章	はじめに	13
	概要	13
	このマニュアルについて	14
	前バージョンとの互換性	15
	動作環境	15
第 2 章	アプリケーションアーキテクチャ	17
	4th Dimension と Mach-O (Mac OS)	17
	Mach-O (マーク・オー) とは?	17
	Mach-O と Mac OS 9	17
	プラグインの互換性	17
	新しいプラグインアーキテクチャ	18
	プラグインの新しい場所	18
	プラグインの内部アーキテクチャ	21
	Mac OS X におけるコマンドラインインタフェース (CLI)	23
	基本情報	23
	コマンドと引数	24
	Mac OS X サービスとして 4D Server を登録する	25
	シンタックスヘルプファイル	26
第 3 章	ユーザモード	29
	ユーザフォームエディタ	29
	エディタの説明	29
	オブジェクトの表示と編集	31
	オブジェクトライブラリの使用	34
	フォーミュラエディタ	35
	インタフェース	35
	コマンドリスト	36
	読み込みと書き出し	36
	XSL 変換を適用する	37
	ODBC データソースを使用した読み込みと書き出し	37
	ODBC データソースとは?	38

ODBCソースからのデータ読み込み	39
ODBCソースへのデータ書き出し	41
クイックレポートエディタ	43
コードの生成	43
標準偏差	45

第4章 デザインモード 47

ログインダイアログボックス	48
パスワードの変更	48
データベースアクセスパラメータの保存	50
環境設定	52
比較表	52
新しい互換性オプション	55
フォームエディタ	56
メモリキャッシュの管理	58
自動クライアント再接続機能を使用する	61
4D Customizer Plusから統合された新機能	62
インタフェース	67
ツールバー	67
メニューの再編成	69
インスペクタ	73
ランタイムエクスプローラ	73
ツールボックス	74
コンパイラ	86
エクスプローラ	86
ゴミ箱ページ	87
コンポーネントページの詳細表示	90
メソッドのプレビューとプロパティの表示	91
エクスプローラでフォルダを使用する	92
はじめに	92
デフォルトフォルダ	94
フォルダやサブフォルダを作成する	94
オブジェクトの削除	96
フォルダとサブフォルダを整理する	97
フォルダ情報	100
4D Insiderのグループを使って作業する	101
メソッドエディタ	101
4Dコードマクロの呼び出し	102
<method>タグ	102
呼び出されたメソッドの実行コンテキスト	102
新しいマクロフォルダ	105

新しいツールバー	106
フォームエディタ	107
パレットとプロパティリスト	107
統合されたオブジェクトバーとツールバー	111
フォームの印刷設定を編集する	115
ズーム	116
ビューの管理	116
バッジを使用する	121
テンプレートとして使用	123
グループ化されたオブジェクトや背景オブジェクトを選択する	124
マグネティックグリッドを使用する	125
ユーザ編集可	126
ダイナミック調整	127
フォームオブジェクト	128
リストボックス	128
スプリッタの管理	140
プラットフォームインタフェース	142
ネイティブな入力制御	144
スペルチェック	146
3ステートチェックボックス	148
新しい3Dボタン	148
その他の変更	156
レコードリストの管理	158
選択モード	159
リスト更新	161
リストの表示	161
サブフォームオプション	165
標準アクション	167
オブジェクトライブラリ	168
オブジェクトライブラリを作成する	168
オブジェクトライブラリを開く	169
オブジェクトライブラリの構築	170
オブジェクトライブラリの表示	172
クライアント/サーバの対応	173
アプリケーションビルダ	173
配布用アプリケーションの新しい名称	174
ダイアログボックスとアプリケーションプロジェクトの構築	174
クライアント/サーバアプリケーションを構築する	176
プラグイン管理	181

シリアル番号の管理	182
クライアントアプリケーションの自動更新	183
Web サービス	184
DOCモードでの公開 (サーバ)	185
Webサービスの利用 (クライアント)	185

第 5 章 統合されたバックアップモジュール..... 1 8 9

はじめに	189
バックアップの管理	190
バックアップの実行	190
バックアップファイルの設定	194
バックアップ設定	199
定期的バックアップの設定	205
自動バックアップ	206
ログファイルの管理	207
はじめに	208
ログファイルの作成	210
ログファイルを中止する	213
ログファイルの解析	213
データの復元	216
障害と診断	216
自動復元	217
手動でバックアップを復元する	220
手動でログを統合する	222
処理の取り消し	223
バックアップジャーナル	224
XMLパラメータファイル	225
4D Backup について	226
互換性	226
サポートされない4D Backupの機能	226
新しいバックアップモジュールのメリット	227

第 6 章 Web サーバ..... 2 2 9

持続的接続 (keep alive 接続)	229
持続的接続を使用する理由	230
keep alive 接続を使用する	230
HTTPS のポート番号を変更する	231
未知の URL のサポート	232
Mac OS X 上で Web サーバを開始する	233
Web サーバの検証	234

第 7 章	ランゲージ	237
	フォーミュラ	239
	SET ALLOWED METHODS	239
	GET ALLOWED METHODS	240
	EDIT FORMULA	241
	ユーザフォーム	242
	ユーザフォーム入門	242
	ユーザフォームとライブラリオブジェクト	243
	ユーザフォームの保存と管理	243
	EDIT FORM	244
	CREATE USER FORM	245
	LIST USER FORMS	246
	DELETE USER FORM	246
	ユーザフォームのエラーコード	247
	階層リスト	247
	SELECT LIST ITEMS BY POSITION (SELECT LIST ITEM) ..	248
	SELECT LIST ITEMS BY REFERENCE	249
	SET LIST PROPERTIES	250
	GET LIST PROPERTIES	251
	SET LIST ITEM PROPERTIES	252
	GET LIST ITEM PROPERTIES	253
	SET LIST ITEM	254
	GET LIST ITEM	254
	List item parent	255
	Selected list items (Selected list item)	255
	Count list items	256
	ユーザ&グループ	257
	新しいコマンド	258
	SET PLUGIN ACCESS	258
	Get plugin access	259
	USERS TO BLOB	259
	BLOB TO USERS	260
	Get default user	261
	変更されたコマンド	262
	CHANGE CURRENT USER (CHANGE ACCESS)	262
	GET USER PROPERTIES	264
	Set user properties	264
	独自のアクセス管理ダイアログボックスを提供する ..	265
	ウインドウ	266
	新しいコマンド	266

Current form window	266
RESIZE FORM WINDOW	267
変更されたコマンド	268
SET WINDOW RECT	268
サイドプッシャー	269
ポップアップウインドウ	271
シートウインドウ	272
フォーム	274
SET FORM SIZE	274
SET FORM HORIZONTAL RESIZING	276
SET FORM VERTICAL RESIZING	277
GET FORM OBJECTS	278
変更されたコマンド	279
INPUT FORM	279
OUTPUT FORM	280
リレート	280
GET FIELD RELATION	281
SET FIELD RELATION	284
GET AUTOMATIC RELATIONS	286
フォームイベント	286
ロールオーバーに関連するフォームイベント	287
3Dボタンに関連するフォームイベント	288
リストボックスに関連するフォームイベント	289
レコードリストに関連するフォームイベント	291
ユーザインタフェース	292
GET TABLE TITLES	293
GET FIELD TITLES	293
SCROLL LINES	294
Pop up menu	295
Focus object	295
入力制御	296
EDIT ITEM	296
印刷	298
サーバ上での印刷	298
SET PRINT OPTION/GET PRINT OPTION (「印刷」テーマ)	299
メッセージ	300
DISPLAY NOTIFICATION	300
データ入力	301
DIALOG	301

システムドキュメント	302
Select document	302
オブジェクトプロパティ	305
Get format	306
SET ENTERABLE	307
SET FORMAT	308
SET RGB COLORS	310
SET COLOR	312
リストボックス	312
“リストボックス” オブジェクトタイプのプログラミング入門	313
新しいコマンド	318
INSERT LISTBOX COLUMN	318
DELETE LISTBOX COLUMN	319
Get number of listbox columns	320
SORT LISTBOX COLUMNS	321
SET LISTBOX COLUMN WIDTH	322
Get listbox column width	322
MOVED LISTBOX COLUMN NUMBER	323
SELECT LISTBOX ROW	323
INSERT LISTBOX ROW	324
DELETE LISTBOX ROW	325
Get number of listbox rows	326
SET LISTBOX ROWS HEIGHT	326
Get listbox rows height	327
MOVED LISTBOX ROW NUMBER	327
GET LISTBOX ARRAYS	328
Get listbox information	329
SHOW LISTBOX GRID	330
SET LISTBOX GRID COLOR	331
SHOW LISTBOX SCROLLBAR	332
4D 環境	332
OPEN 4D PREFERENCES	332
システム環境	339
SET ENVIRONMENT VARIABLE	339
LAUNCH EXTERNAL PROCESS	339
セレクション	341
GET HIGHLIGHTED RECORDS	342
Displayed line number	343
HIGHLIGHT RECORDS	344
MODIFY SELECTION、DISPLAY SELECTION	345
GOTO SELECTED RECORD	346

Webサーバ	346
PROCESS HTML TAGS	347
GET HTTP BODY	348
SEND HTTP RAW DATA	349
GET WEB FORM VARIABLES	352
CGIのサポート	352
SET CGI EXECUTABLE	352
配列	353
Count in array	354
APPEND TO ARRAY	354
通信	355
GET SERIAL PORT MAPPING	355
バックアップ	355
BACKUP	356
RESTORE	356
GET BACKUP INFORMATION	357
GET RESTORE INFORMATION	358
「On Backup Startup」データベースメソッド	359
「On backup shutdown」データベースメソッド	359
XML	360
4D 2003におけるコマンド名の変更	360
SAX標準のサポート	361
SAX SET XML OPTIONS	362
SAX ADD XML DOCTYPE	363
SAX ADD XML COMMENT	363
SAX OPEN XML ELEMENT	364
SAX OPEN XML ELEMENT ARRAYS	365
SAX CLOSE XML ELEMENT	366
SAX ADD XML ELEMENT VALUE	366
SAX ADD XML CDATA	367
SAX ADD PROCESSING INSTRUCTION	368
SAX Get XML node	369
SAX GET XML DOCUMENT VALUES	370
SAX GET XML COMMENT	370
SAX GET XML PROCESSING INSTRUCTION	371
SAX GET XML ELEMENT	371
SAX GET XML ELEMENT VALUE	372
SAX GET XML CDATA	373
SAX GET XML ENTITY	374
DOMコマンドを用いてXMLを設定する	374
DOM SET XML OPTIONS	376

DOM Create XML Ref	377
DOM SET XML ATTRIBUTE	378
DOM Create XML element	379
DOM SET XML ELEMENT NAME	381
DOM SET XML ELEMENT VALUE	382
DOM REMOVE XML ELEMENT	382
DOM EXPORT TO FILE	383
DOM EXPORT TO VAR	384
DOM Find XML element	384
変更されたコマンド	385
DOM GET XML ELEMENT VALUE	385
XML エラーコード	385
XSL Transformations の管理	386
APPLY XSLT TRANSFORMATION	387
SET XSLT PARAMETER	389
GET XSLT ERROR	390
Webサーバ (クライアント側)	390
SET WEB SERVICE OPTION	390
外部データソース	392
ODBC LOGIN	393
ODBC LOGOUT	394
ODBC SET OPTION	395
ODBC GET OPTION	395
ODBC EXECUTE	396
ODBC End selection	398
ODBC LOAD RECORD	398
ODBC CANCEL LOAD	399
ODBC SET PARAMETER	400
ODBC GET LAST ERROR	402
ODBC IMPORT	403
ODBC EXPORT	404
ツール	406
BUILD APPLICATION	406
ENCODE	407
DECODE	407
SPELL CHECKING	408
SET DICTIONARY	408

4th Dimension 2004へようこそ！4th Dimension 2004は4th Dimensionおよび4D Server開発環境の最新バージョンです。

概要

4th Dimension 2004には数々の新機能が追加され、プログラム機能の大部分に対して最適化が行われています。これらの新機能は主に5つに分類されます。

4th Dimension 2004では、Windows XPおよびMac OS Xをはじめとする最新世代のOSとの統合が大幅に計られています。統合強化された機能としては、より高度なインタフェース仕様のサポート、メモリ管理機能、Mac OS X Mach-Oのサポートなどがあります。この結果として、4th Dimension 2004製品は、Apple社のMac OS 9.x、Windows 98、Windows Millenniumオペレーティングシステムとの互換性を持たなくなりました。

現在最も利用されているテクノロジーに対してオープンであり続けるため、4th Dimension 2004ではさまざまな対応が引き続き行われています。高度なWebサービスやCGIサポートに加えて、4th Dimension 2004は新しいXMLコマンドやXSLサポート、ODBC形式でのデータ読み込みと書き出し、新しいハイレベルODBCコマンドによって数多くの可能性を提供し、外部のシステムやアプリケーションに対する公開性をより一層高めています。

重点が置かれているもう一つの開発分野として、柔軟で効率的なバックアップソリューションの統合があります。今後、4th Dimensionにはデータバックアップと復旧用のモジュール一式が含まれます。また、このモジュールはログファイルの追加統合に対応します。

開発環境もまた大幅に向上しています。さらに機能的になったツールバー、フォームエディタの変更（ズーム、マゲネティズム、ビューなど）、「エクスプローラ」におけるオブジェクトフォルダの設定、新しいフォームイベント、約100種類の新しいランゲージコマンドは、4th Dimension開発者の作業環境対して行われた数々の機能向上のほんの一部です。

4th Dimension 2004では、新しい“リストボックス”や“ベベルボタン”フォームオブジェクト、機能向上したリスト管理（サブフォーム、出力フォームなど）、ユーザ選択によるフォーム変更機能やオブジェクトライブラリーへのアクセス機能など、新しいツールがいくつか提供されています。これらのツールを用いて、最終アプリケーションのユーザインタフェースをより一層充実させることができます。

最後に、開発者向けに新しい配備用ツールが提供されています。柔軟で効率的な統合バックアップシステム（バックアップ、復旧、ログファイルの追加統合の管理に対応）、全体のカスタマイズやプログラムによる制御が可能なクライアント/サーバアプリケーションビルダ、クライアント側の自動更新メカニズムは、特に注目すべきツールです。

このマニュアルについて

本マニュアルは、バージョン2004の4th Dimensionおよび4D Serverに導入された新しい機能と変更点についてすべて詳しく説明しています。マニュアルは、以下の章に分かれています。

- **アプリケーションアーキテクチャ**：この章では、4th Dimension 2004アプリケーションのファイルとプラグインの内部アーキテクチャに関する変更点について説明します。
- **ユーザモード**：この章では、4th Dimensionの「ユーザ」モードにおける新機能について説明します。新しい「フォーミュラ」エディタ、ODBC形式でのデータ読み込み／書き出し、ならびに「クイックレポート」エディタの変更点について説明しています。
- **デザインモード**：この章では、「デザイン」モードにおける新しい機能や変更点について説明します。全般的な新しい「環境設定」、ツールボックスとしてまとめられた最新のエディタ、ODBCテーブルのクローニング機能、「フォーム」エディタの変更点、新しいフォームオブジェクト、拡張されたWebサービスサポートなどについて説明しています。
- **統合バックアップモジュール**：この章では、新しい統合バックアップモジュールについて説明します。このモジュールは、データの保存と復元、およびログファイル管理に使用します。
- **Webサーバ**：この章では、4th Dimension 2004のWebサーバにおける変更点について説明し、特にCGIの使用に関する向上とHTTP 1.1のサポートについて詳しく述べています。
- **ランゲージ**：この章では、4th Dimension 2004のプログラミング言語に関する新機能および変更点（新しいコマンドテーマ、コマンド、データベースメソッドや変更されたコマンド）について説明します。

- 4th Dimension 2004 プロダクトラインのプラグイン（特に4D Pack、4D Write、4D View、4D Draw）もまた変更されています。これらの変更点については、『プラグインアップグレード』マニュアルで説明しています。

4th Dimension 2004には、ODBCプラグインが搭載されています。このプラグインは、ODBC互換アプリケーションとの任意の接続タイプ設定に使用できるローレベルコマンドを含みます。このプラグインについては、別のドキュメントで説明しています。

前バージョンとの互換性

4th Dimension バージョン 6.7.x、6.8.x、2003 または 4D Server を使用して作成されたデータベースは、4th Dimension 2004 のアプリケーション（ストラクチャファイルおよびデータファイル）と完全に互換します。

しかし、4th Dimension 2004 でデータベースを開くとストラクチャファイルとデータファイルが変換されるため、以前のバージョンの 4th Dimension でこれらのデータベースを開くことができなくなるため注意が必要です。

以前のバージョンの 4D で作成されたデータベースを 4th Dimension 2004 で開くと、2種類のダイアログボックスが連続して表示され、ストラクチャファイルとデータファイルが変換されることを知らせます。



動作環境

最新世代の OS を利用した 4th Dimension の統合開発を継続することにより、これらのシステムの従来のバージョンをサポートできなくなりました。この結果、4th Dimension 2004 を利用できる環境は、次の通りになります。

- Macintosh：Mac OS 9.x はサポートされなくなります。Macintosh の最小バージョンは、バージョン 10.1 になります。

■ **Windows** : Windows 98 SEならびにWindows Milleniumオペレーティングシステムはサポートされなくなります。Windowsの最小バージョンは、Windows 2000となります。

4th Dimension 2004 プロダクトラインのアプリケーションには、最低でも次の動作環境が必要です。

	Windows	Mac OS X
コンピュータ	Pentium IIプロセッサ搭載のPC互換機	G3
OS	Windows 2000、Windows XP	バージョン10.2.8
メモリ	256MB	
画面解像度	800×600ピクセル	

注：4Dデータベースでプラグインを使用する場合、最適なパフォーマンスでアプリケーション（4th Dimension、4D Client、4D Runtime、実行形式アプリケーションなど）を実行するために、そのマシン上でさらに2MB程度の利用可能なRAMが必要となります。この値はアプリケーションで使用するプラグインに応じて変わります。

4th Dimension と Mach-O (Mac OS)

Mac OS 版 4th Dimension 2004 のカーネルアプリケーション (4th Dimension、4D Server、4D Client、4D Runtime) ならびに主な 4D プラグインは、Mach-O フォーマットへ移植されました。

Mach-O (マーク・オー) とは？

Mach-O (Mach Object-ファイルフォーマット) は、Mac OS X のネイティブなアプリケーション実行形式です。このフォーマットで提供されるアプリケーションは、アクセス制御やバックオフィスサービスの点に関し、より緊密にシステムへ統合されています。また、Mac OS X 上での全体的なパフォーマンスも向上します。

Mach-O フォーマットにより、4D アプリケーションで次の事柄が実現します。

- カレントユーザがルートユーザではない場合に、TCP ポート 80 (標準) で Web サーバを開始する。
- 4D カーネル (4th Dimension、4D Server、4D Client、4D Runtime) をサービスとして登録し、データベースを自動的にオープンする。

Mach-O と Mac OS 9

Mach-O フォーマットは Mac OS 9 との互換性がありません。

このため、4th Dimension 2004 パッケージ (バンドル) は Mac OS 9 や Mac OS X Classic モードで動作しません。

プラグインの互換性

Carbon 形式ではないプラグイン (旧世代) は、Mac OS 版の 4th Dimension 2004 と互換しくなくなります。4th Dimension でプラグインをロードできるようにするには、Mach-O または CFM フォーマットでプラグインを Carbon アプリケーションに変更しなければなりません。

注：CFM（Code Fragment Manager）は“従来の”プラグインライブラリです。Carbon形式に変更されると、これらのプラグインはMac OS Xとの互換性を維持します。

4Dプラグインはすべて、Mach-Oフォーマットへ移植されました。

- 4D View
- 4D Write
- 4D Draw
- 4D for OCI
- 4D ODBC 2004
- 4D Open for 4D
- 4D Internet Commands
- 4D Pack

次の4Dプラグインは4th Dimension 2004で使用できません。

- 4D Calc
- 4D List
- 4D ODBC

4D 2003（またはそれ以前）のプラグインは、4th Dimension 2004との互換性がありません。4th Dimension 2004では、バージョン2004の4Dプラグインだけを使用することができます。この制約はサードパーティのプラグインに影響しません。

新しいプラグインアーキテクチャ

4th Dimension 2004のプラグインアーキテクチャでは、プラグインファイルの場所、ならびにプラグインの内部構造が変更されています。

プラグインの新しい場所

4th Dimension 2004において、プラグインは2種類の場所に配置することができます。

- データベースストラクチャファイルと同じ階層
- 4th Dimensionアプリケーションと同じ階層

注：4th Dimension 2004 は、アクティブな 4D フォルダ内に配置されたプラグインをロードしません。

ただし、プラグインのプレファレンスは、従来通りアクティブな 4D フォルダ内に保存されますので注意してください。

また、2種類のフォルダを使用してプラグインを保存することができます。

■「プラグイン (PlugIns)」フォルダ

■「Mac4DX」または「Win4DX」フォルダ（データベースのストラクチャファイルと同じ階層に限定）

次の節では、プラグインのインストールについて説明します。

4D アプリケーションと同じ階層に配置されたプラグイン

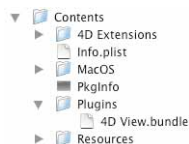
4D アプリケーションと同じ階層に配置されたプラグインは、4D アプリケーションを使用して開かれたすべてのデータベースから利用できます。

この場所に配置した場合、「PlugIns」フォルダだけが参照されます。「Win4DX」や「Mac4DX」フォルダをこの場所に配置することはできません。

このフォルダにコピーした Mac OS プラグインは、そのフォーマットが何であれ（Mach-O または CFM）、必ずパッケージとして提供しなくてはなりません。パッケージとして提供しないプラグインは、「Win4DX」や「Mac4DX」フォルダにのみ配置することができます。

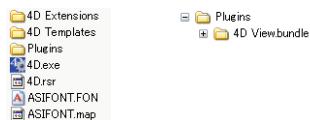
Windows において、このフォルダにコピーされたプラグインは、Macintosh のパッケージ構造を再現しなければなりません（後述の「プラグインの内部アーキテクチャ」の節を参照）。

■ Mac OS において、「PlugIns」フォルダはパッケージの「Contents」フォルダ内に配置されます。



注：Mac OS X 上でパッケージの内容を表示するには、Control キーを押したままアイコンをクリックし、次にコンテキストメニューから「パッケージの内容を表示する」コマンドを選択してください。

- Windowsにおいて、「PlugIns」フォルダは4Dアプリケーションと同じ階層に配置されます



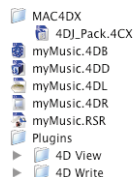
データベースストラクチャファイルと同じ階層に配置されたプラグイン

データベースストラクチャファイルと同じ階層に配置されたプラグインは、そのデータベースでしか利用できません。

この場所に配置した場合、「PlugIns」フォルダまたは「Win4DX」や「Mac4DX」フォルダのいずれかを使用することができます。

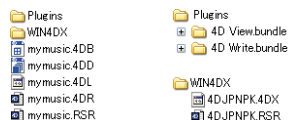
- Mac OS上では、「PlugIns」フォルダ内にパッケージとしてプラグインを納め、他のプラグイン（標準のアーキテクチャ）は「Mac4DX」フォルダに納めなくてはなりません。

もちろん、「Mac4DX」フォルダにプラグインをパッケージとして配置することは可能です。



- Windows上では、Macintoshのパッケージ構造に準じて「PlugIns」フォルダ内にプラグインを納め（後述の「プラグインの内部アーキテクチャ」の節を参照）、「Win4DX」フォルダ内に他のプラグイン（標準のアーキテクチャ）を納めなくてはなりません。

もちろん、「Win4DX」フォルダにパッケージ構造タイプのプラグインを配置することは可能です。



ロード優先順位

複数の異なる場所に同じプラグインが存在する場合、4th Dimensionは次のルールを適用します。

- 異なるレベルに同じプラグインが2つ存在する場合（データベースストラクチャとアプリケーション）、4th Dimension はストラクチャレベルに置かれたプラグインだけをロードします。
- 同じレベル（データベースストラクチャまたはアプリケーション）の異なる2つの場所に同じプラグインが2つ存在する場合（「Win4DX/Mac4DX」と「PlugIns」）、4th Dimension はエラーメッセージを表示し、データベースは起動されません。

注：コンパイルされて4D Engine が組み込まれたアプリケーションの場合、同じプラグインが複数個存在するとアプリケーションを開くことができません。

プラグインの内部アーキテクチャ

“新世代”のプラグインを「PlugIns」フォルダに納めて使用する場合には、Mac OS のパッケージアーキテクチャを遵守しなくてはなりません。Windows 上においても、このアーキテクチャを推奨します。

注：このアーキテクチャに準拠していないプラグインは、必ずデータベースストラクチャファイルと同じ階層にある「Win4DX」や「Mac4DX」フォルダ内に配置してください。

プラグインのフォルダ名には、拡張子“.bundle”を必ず付加してください。各プラグインのバンドルには、ひとつのランゲージだけを納めます。

スタンドアロン

スタンドアロンの4th Dimension を使用する場合、プラグインの内部アーキテクチャを次のように設定しなくてはなりません。

- Mac OS の場合：

```
MyPlugIn.bundle/  
  Contents/  
    Info.plist  
    MacOS/  
      MyPlugIn (Mach-O または CFM コード)  
    PkgInfo  
    Resources/  
      MyPlugIn.rsrc (リソース)
```

■ Windows の場合：

```
MyPlugIn.bundle/  
  Contents/  
    Windows/  
      MyPlugIn.4DX  
      MyPlugIn.RSR
```

クライアント / サーバ

クロスプラットフォーム対応のクライアント / サーバ構成のもとで作業を行う場合、Windows および Mac OS フォルダ（「Resources」フォルダを納める）を同じ「.bundle」フォルダ内に配置します。通常、クロスプラットフォーム対応のプラグインは次のような構造になります。

```
MyPlugIn.bundle/  
  Contents/  
    Info.plist  
    Mac OS/  
      MyPlugIn (Mach-O または CFM コード)  
    PkgInfo  
    Resources/  
      MyPlugIn.rsrc (リソース)  
    Windows/  
      MyPlugIn.4DX  
      MyPlugIn.RSR
```

注：Info.plist と PkgInfo ファイルは、Mac OS 上で実行するプラグインでのみ使用されま
す。しかし、Windows 版 4D Server 上でクロスプラットフォームのバンドルフォルダを
設定する場合、これらのファイルはオプションとなります。

クライアント / サーバプラグインのダウンロードのメカニズム

クライアントマシン（4D Client）から 4D Server へ接続する際、サーバ上にインストール
されているプラグインがクライアントマシン（4D Client）へ送信されます。後述の表に示
す通り、サーバ構成に応じて、ダウンロード後のプラグインはクライアントマシン上の
「PlugIns」フォルダや、「Win4DX」または「Mac4DX」フォルダ内に配置されます。

- 「PlugIns」フォルダや「Win4DX/Mac4DX」フォルダがサーバ上のデータベースストラクチャファイルと同じ階層にインストールされている場合

OS	クライアントマシン上にダウンロードされたフォルダの場所
Mac OS X	{Disk}:Library:Application support:4D:データベース名:PlugIns (またはMac4DXかWin4DX)
Windows 2000 Windows XP	{Disk}:¥Documentsおよび Settings¥ユーザ名¥Application Data¥4D¥データベース名¥PlugIns (またはMac4DXかWin4DX)

- 「PlugIns」フォルダがサーバマシン上の4D Serverアプリケーションと同じ階層にインストールされている場合

OS	クライアントマシン上にダウンロードされたフォルダの場所
Mac OS X	{Disk}:Library:Application support:4D:データベース名:4D:PlugIns
Windows 2000 Windows XP	{Disk}:¥Documentsおよび Settings¥ユーザ名¥Application Data¥4D¥データベース名¥4D¥PlugIns

注：4D Server 2004では、ダウンロードフォルダ内に配置された項目（フォルダとファイル）の階層構造がクライアントマシン上でも保持され、再構築されます。

クライアントマシンのOSに対応する項目だけがダウンロードされます。

注：また、4D Serverにおいて内部的に必要となるため、「Cache」、「Server」、「Database」という名称のフォルダが追加されます。

Mac OS Xにおけるコマンドラインインタフェース (CLI)

Mac OS Xの「ターミナル」でコマンドラインを使用し、4th Dimensionアプリケーション (4th Dimension、4D Server、4D Client) を作動させることができます。この機能により、データベースのオープンやクローズをリモートから行えるようになり、特にWebサーバを管理する上で役立ちます。

基本情報

Mac OS Xの「ターミナル」を使用して大部分の4th Dimensionコマンドを実行できるようにするには、まず初めにパッケージ内のアプリケーションが格納されたフォルダへアクセスしなければなりません (Contents/Mac OSパス)。例えば、4th Dimensionパッケージがフォルダ「MyFolder」内に配置されている場合、次のようなコマンドラインを記述してください。

```
/MyFolder/4th\ Dimension.app/Contents/Mac OS/4th\ Dimension
```

注：「ターミナル」上で、アプリケーションアイコンをドラッグ&ドロップし、コマンドラインを挿入することもできます。

コマンドラインの最後に“&”記号を記述することをお勧めします。これにより、そのアプリケーションはバックグラウンドタスクとして異なるプロセスで実行されます。例えば、次のように記述します。

```
/MyFolder/4th\ Dimension.app/Contents/Mac OS/4th\ Dimension&
```

コマンドと引数

以下の節では、4th Dimension アプリケーションによって処理されるコマンドラインと、その引数について説明します。

■ アプリケーションの起動

シンタックス：パス/アプリケーション名

例：

```
4th\ Dimension.app/Contents/Mac OS/4th\ Dimension&
```

このコマンドの動作は、4th Dimension アプリケーションをダブルクリックした場合と同じです。アプリケーションが実行され、データベースを選択するダイアログボックスが表示されます。

■ ストラクチャファイルを伴うアプリケーションの起動

シンタックス：パス/アプリケーション名 ストラクチャパス

例：

```
4th\ Dimension.app/Contents/Mac OS/4th\  
Dimension/Users/Roger/Bases/MyBase/MyBase.4DB
```

このコマンドはアプリケーションを起動し、カレントデータファイルとセットになっているストラクチャファイルを開きます。ダイアログボックスは表示されません。

■ ストラクチャファイルとデータファイルを伴うアプリケーションの起動

シンタックス：パス/アプリケーション名 -dデータパス ストラクチャパス

例：

```
4th\ Dimension.app/Contents/Mac OS/4th\ Dimension-d  
/Users/Roger/Bases/MyBase/MyBase.4DD/Users/Roger/Bases/MyBase/MyBase.4DB
```

このコマンドはアプリケーションを起動し、選択したデータファイルとセットになっているストラクチャファイルを開きます。ダイアログボックスは表示されません。

■ アプリケーションの終了

シンタックス：kill-s INT プロセス_ID

例：

Kill-s INT 323

このコマンドの動作は、「ファイル」メニューから「終了」コマンドを選択した場合と同じです。各プロセスをひとつずつ終了してキャッシュを保存し、アプリケーションを終了します。

注：4th Dimension アプリケーションのプロセス番号（プロセス_ID）は、ps または top などのコマンドを使用して取得することができます。

■ ヘルプ

シンタックス：パス/アプリケーション名 -h

例：

4th\ Dimension.app/Contents/Mac OS/4th\ Dimension-h

このコマンドは、Mac OS X のターミナル上で 4th Dimension とともに使用できるコマンドの概要を表示します。

■ カレントセッション以外のユーザセッションにおけるアプリケーションの起動

シンタックス：パス/アプリケーション名 -u ユーザ名 -g グループ名

例：

4th\ Dimension.app/Contents/Mac OS/4th\ Dimension-u john-g acctnt

このコマンドは、指定されたユーザセッションを使用して強制的にアプリケーションを起動します（デフォルトとしてカレントユーザセッションを使用）。これにより、一時的ルートのセッションを用いた Web サーバの起動後に使用されるセッションを指定することができます（後述の「Mac OS X 上で Web サーバを起動する」の節を参照）。

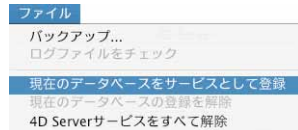
Mac OS X サービスとして 4D Server を登録する

Mac OS X において、サービスとして 4D Server 2004 を登録する方法が簡単になりました。

注：Windows 上で 4D Server をサービスとして登録するメカニズムは変更されていません。

サービスとして登録された4D Serverは、ユーザセッションのオープン前であっても、マシンの開始時に自動的に起動されます（カレントデータベースを使用）。この方式により、マシンの再起動が必要となる事態が発生しても、必ず4D Serverデータベースを利用できるようになります。保守管理はリモートで行われます。

Windowsの場合と同様に、Mac OS上で4D Serverをサービスとして登録あるいは登録解除するには、サーバ側の「ファイル」メニューから特定のコマンドを使用します。



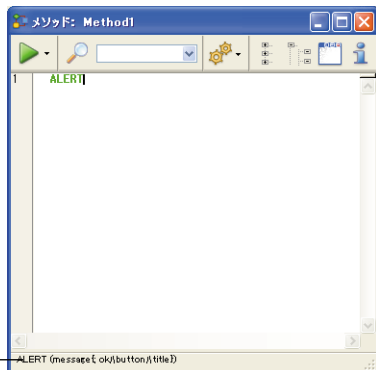
- **現在のデータベースをサービスとして登録：**カレントデータベースをサービスとして登録します。次回そのマシンを起動すると、4D Serverが自動的に起動され、カレントデータベースが開かれます。

注：サービスとして登録するストラクチャファイル名には日本語は使えません。また、4D Serverアプリケーションおよびストラクチャファイルのフルパスにあるフォルダ名に日本語を使うこともできません。

- **現在のデータベースの登録を解除：**カレントデータベースのサービスとしての登録を解除します。データベースがサービスとして登録されていない場合、このコマンドはグレー表示されます。
- **すべての4D Serverサービスの登録を解除：**そのマシン上で宣言されたすべての4D Serverサービスの登録を解除します。アクティブな4D Serverサービスが存在しない場合、このコマンドはグレー表示されます。

シンタックスヘルプファイル

4th Dimension 2004では、シンタックスヘルプファイルが変更されています。開発環境にこのファイルをインストールすると、メソッドエディタ上に4Dコマンドの構文が表示され、先行入力されます。



シンタックスヘルプ

また、「エクスプローラ」(Windows)には、各コマンドの追加情報も表示されます。

以前のバージョンの4th Dimensionにおいて、このファイルには「4D Help (Mac OS)」または「4D Help.rsr」という名前が付けられ、アクティブな4Dフォルダに格納する必要がありました。

使用方法を明確にし、その更新を容易にするため、このファイル名は両プラットフォームともに「4D Syntax.rsr」に変更されており、4Dアプリケーションの「4D Extensions」フォルダに配置しなくてはなりません。

4th Dimension 2004では、「ユーザ」モードにいくつかの新しい機能が追加されています。

- 新しいユーザフォームエディタ
- フォーミュラエディタの変更
- ODBCソースからのデータ読み込み／ODBCソースへのデータ書き出し
- 「クイックレポート」エディタ：自動コード生成と新しい「標準偏差」機能
- 4th Dimension インタフェースと「ユーザ」モードに関する各種変更については、第4章「デザインモード」の「インタフェース」の節で説明しています。

ユーザフォームエディタ

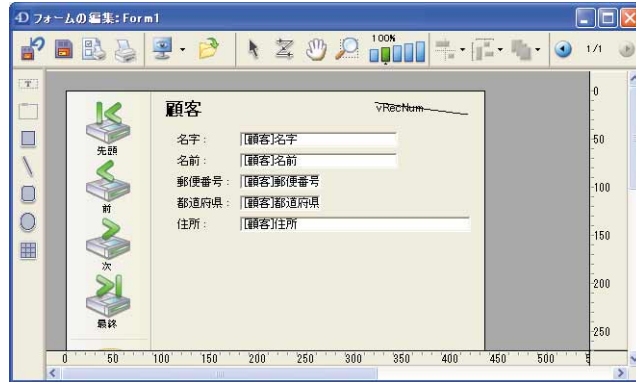
4th Dimension 2004では、例えばカスタマイズしたレポートを作成するために、開発者がユーザに対してデータベースフォームの変更を許可することができます。このフォームを保存すると、“ユーザフォーム”として他の4th Dimension フォームと同じように利用することができます。

このシステムの設定方法については、後述の「ユーザフォーム」の節で説明しています。次の節では、「ユーザフォーム」エディタの操作方法を説明します。このエディタ上で、ユーザはフォームのカスタマイズを行うことができます。

エディタの説明

EDIT FORM コマンドを実行し、その引数として渡したフォームに対し「ユーザ編集可」プロパティが選択されている場合、「ユーザフォーム」エディタが表示されます。

次のようなエディタが表示されます。



このエディタは、4th Dimension 標準の「フォーム」エディタインタフェース、特に（ツールとオブジェクトの）2つが統合されたツールバーの概念を利用しています。さらに、このエディタ特定の機能もあります。

エディタが表示された時は、メニューバーの「編集」メニューだけが利用できます。さらに、簡略化された「プロパティリスト」とコンテキストメニューも表示することができます。

オブジェクトの選択や移動のメカニズムは（許可されている場合）、4th Dimension 標準の「フォーム」エディタと同じです。

ツールバー

ツールバーには次のアイテムがあります。



作成時の設定へ戻す：「デザイン」モードで定義されたオリジナルのフォームに戻します（ユーザフォームで行われた変更はすべて破棄）。このボタンをクリックすると警告ダイアログボックスが表示され、操作のキャンセルや確定を行うことができます。



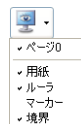
保存：ユーザフォームに対する変更を保存します。



印刷設定および印刷：これら2つのコマンドを使用し、ユーザフォームの印刷設定や印刷を行うことができます。



表示：このボタンのポップアップメニューを使用して、エディタ上の選択項目を表示したり、または隠すことができます。



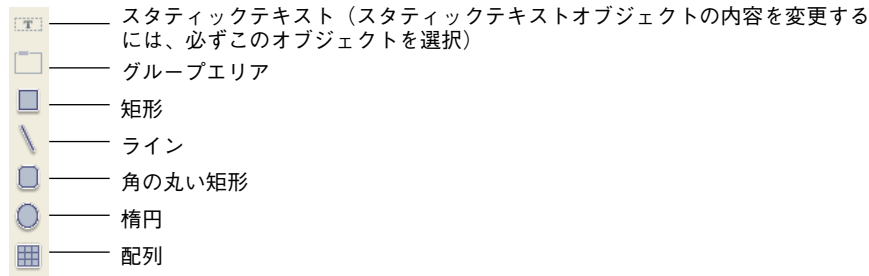


ライブラリを開く：フォームで使用できるオブジェクトライブラリをロードすることができます（該当する場合）。詳細については、後述の「オブジェクトライブラリへのアクセス」の節を参照してください。

その他のツールは、4th Dimension 標準の「フォーム」エディタのツールバーと同じです。詳細については、後述の「ツールバー」の節を参照してください。

オブジェクトバー

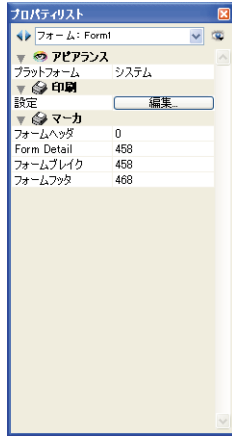
「フォーム」エディタのオブジェクトバーを使用し、各種グラフィックオブジェクトをフォームに追加することができます。



フォーム上でオブジェクトを作成するには、オブジェクトバーからオブジェクトを選択し、それをフォーム上に描画します。これらのオブジェクトは「デザイン」モードで使用されるものと同じです。

オブジェクトの表示と編集

簡略版のプロパティリストを使用して、ユーザはフォーム上にあるオブジェクトのプロパティの表示や編集を行うことができます。このリストを表示するには、オブジェクトをダブルクリックするか、エディタのコンテキストメニュー（Windowsでは右クリック、Mac OSではControl+クリック）から「プロパティリスト」コマンドを選択します。

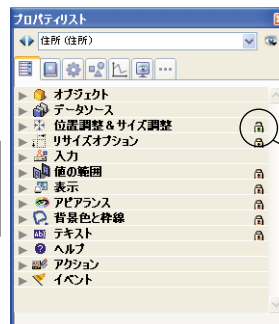


「プロパティリスト」には、ユーザ編集可能なプロパティだけが表示されます。

- ユーザが追加したグラフィックタイプのオブジェクトについては、そのオブジェクトタイプに関係するすべてのプロパティを使用できます。
- オリジナルフォームのオブジェクトに関し、あるプロパティを「ユーザフォーム」エディタ上で編集できるようにするには、次の2つの条件を満たさなくてはなりません。
 - そのプロパティは、オブジェクトやフォームに対して編集可と宣言される一連のプロパティに属していなければなりません。例えば、フォームイベントやドラッグドロップの管理に関連するプロパティは、ユーザ編集可になることは決してありません。「デザイン」モードの「フォーム」エディタにおいて、変更可能なプロパティテーマは「プロパティリスト」の南京錠アイコンを使って区別されています（後述の「ユーザ編集可」の節を参照）。
 - そのオブジェクトまたはフォームに対して、開発者がそのプロパティテーマを編集可であると明示的に宣言しておかなければなりません。これを行うには、開発者は「デザイン」モードの「プロパティリスト」で南京錠アイコンをクリックし、そのロックを解除しなければなりません。

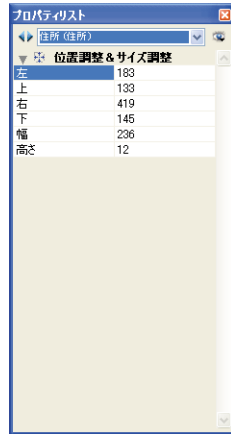
「デザイン」モードのプロパティリスト

編集可のテーマ



ロック解除されたテーマ

ロック解除されたプロパティは、ユーザが「ユーザフォーム」エディタ上で使用することができます。



ロック解除することにより、ユーザフォームにおける他の動作にも影響を与えます。例えば、“位置調整&サイズ調整” テーマのロックが解除されている場合、ユーザはマウスやキーボードを用いてそのオブジェクトを移動することができます。

「編集」メニューコマンド

「ユーザ」モードで「ユーザフォーム」エディタを表示すると、「編集」メニューだけがアクティブになっています。カスタマイズしたインタフェースの場合（「カスタム」メニュー）、「編集」メニュー内のコマンドが標準アクションで管理されていれば、「編集」メニューを使用することができます。

編集用の全コマンド（カット、コピー、ペースト、クリア、すべてを選択）は、ユーザ追加のオブジェクトに対する制約を受けずに動作します。

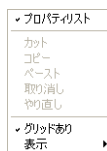
一方、明確にするために、ユーザはオリジナルフォームのオブジェクトを削除することはできません。

「カット」や「クリア」コマンド、または「Del」や「Backspace」キーを使用すると警告ダイアログボックスが表示され、このオブジェクトに対しその動作が許可されていないことをユーザに知らせます。

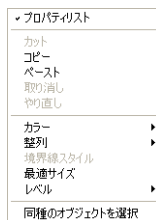
コンテキストメニュー

「ユーザフォーム」エディタのコンテキストメニューを利用して、複数のコマンドを使用することができます。このメニューの内容を制御することはできません。

- ユーザが任意のオブジェクトの外側をクリックすると、フォームエディタ標準の表示コマンドを使用することができます（プロパティリスト表示、マグネティックグリッドのアクティブ化、エディタツールの表示）。



- ユーザがオブジェクト上でクリックすると、グラフィック関連のコマンドを使用することができます（カラー、整列など）。



オリジナルフォームに属するオブジェクトについては、開発者が「デザイン」モードで指定した設定に従い、これらのコマンドの利用が制限されます。

オブジェクトライブラリの使用

編集中の各ユーザフォームでオブジェクトライブラリを利用することができます。

注：オブジェクトライブラリに関する詳細は、後述の「オブジェクトライブラリ」の節を参照してください。

開発者は **EDIT FORM** コマンドの最後の引数を用いて、フォームにライブラリを関連付けすることができます。この引数が有効であれば、ライブラリが表示され、ユーザフォームで使用することができます。

ライブラリを開くには、エディタツールバーの対応するアイコンをクリックします。



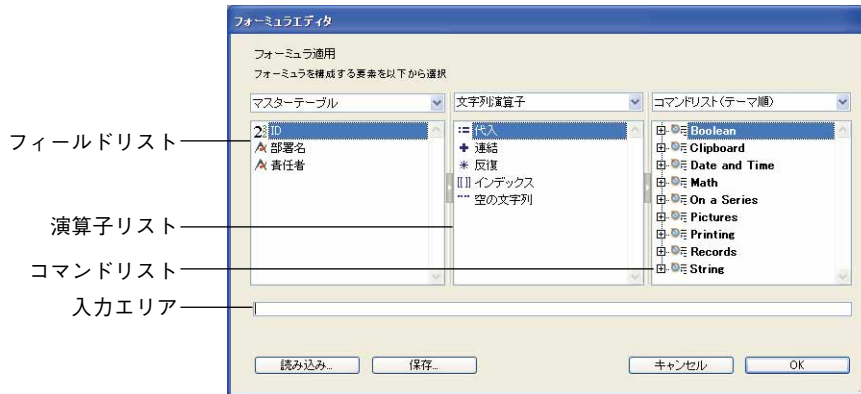
すると、ライブラリウィンドウが表示されます。ユーザは、ライブラリからオブジェクトをドラッグ&ドロップしたり、コピーペーストして、フォームにオブジェクトを追加することができます。オブジェクトは、設定されたすべてのプロパティや自動アクションとともに追加されます（詳細については、後述の「ユーザフォームとライブラリオブジェクト」の節を参照）。

フォーミュラエディタ

4th Dimension 2004では、フォーミュラエディタが変更されています。インタフェースおよび使用可能なコマンドリストに関する変更が行われています。

インタフェース

フォーミュラエディタのインタフェースが変更され、より直感的かつ論理的にフォーミュラを記述できるようになりました。次のようなエディタが表示されます。



3つのリスト内の項目を次々に選択して、フォーミュラを入力できるようになりました。各リストの内容は、リストの下側にあるメニューを用いて編集することができます。リストのサイズはそれぞれ変更可能です。

さらに、入力エリアではドラッグ&ドロップがサポートされる点に注意してください。

- フィールドリストを使用して、フォーミュラを適用するフィールドを選択することができます。以前のバージョンの4th Dimensionと同様に、リストメニューを用いて、カレントテーブルやリレートテーブル、すべてのテーブルのフィールドを表示できます。
- 演算子リストを使用し、フォーミュラで使う演算子を選択することができます。演算子はテーマ別に分類され、リストメニューを用いてアクセスすることができます。



データや処理のタイプに対応する使用可能な演算子がすべて、テーマ別に表示されます。例えば、代入演算子である「:=」はすべてのデータタイプに対して使用できます。次のような新しい演算子を使用することができます。

- Empty string（空の文字列）：2つの引用符""（文字列演算子）を挿入
- Integer division（整数による除算）：¥（数値および時間演算子）
- Exponentiation（指数）：^（数値演算子）
- Blank date（空の日付）：!00/00/00!を挿入（日付演算子）
- Blank hour（空の時間）：?00:00:00?を挿入（時間演算子）
- &（ANDロジック）と|（ORロジック）（論理演算子）
- コマンドリストには、ユーザフォーミュラや開発者が許可したすべてのメソッド内で使用できるコマンドが納められています（後述の節を参照）。以前のバージョンの4th Dimensionと同様に、リストメニューを用いて、テーマ順やアルファベット順でコマンドを表示することができます。

コマンドリスト

4th Dimension 2004では、フォーミュラ上で使用できる関数だけがコマンドリストに表示されます。また、フォーミュラのテキスト内ではこれらの関数しか受け入れられず、認可されていないコマンド名をユーザが入力すると、フォーミュラは実行されません。また、特別に許可されたプロジェクトメソッドだけを使用できます。これにより、データのセキュリティが強化されます。

プロジェクトメソッドへのアクセスを可能にしたい場合は、それを明示的に許可しなくてはなりません。同様に、他のコマンドやプラグインコマンドへのアクセスを許可したい場合、それらのコマンドをプロジェクトメソッド内に挿入し、エディタ上で認可しなくてはなりません。これを行うには、「デザイン」モードにおいて新しいフォーミュラ管理コマンドを使用します。この件に関する詳細は、後述する「フォーミュラ」の節を参照してください。

注：変換後のデータベースでは、この新しい方法により、独自のメソッドを利用するクイックレポートの処理が妨げられるおそれがあります。この場合、そのメソッドを認可する必要があります。

読み込みと書き出し

4th Dimension 2004には、データの読み込みと書き出しに関する新しい機能が追加されています。

- XML フォーマットでのデータ書き出し時に、XSL 変換を直接適用できるようになります。
- 4th Dimension 2004 は ODBC ソースからのデータ読み込みや、データ書き出しをサポートします。

XSL 変換を適用する

データ書き出し用のダイアログボックスにおいて、XML フォーマットでのデータ書き出し時に、新しい2つのオプション「XSL 変換を適用」と「XSL ファイルへのリンク挿入」が提供されます。



これらのオプションは、XSL ファイルが設定された場合にのみ使用できます。「XSL 変換を適用」オプションを選択すると、4th Dimension は書き出し時に XSL データの変換を指定された XSL ドキュメント上で直接実行します。

「XSL ファイルへのリンク挿入」オプションを選択すると、4th Dimension は指定された XSL への参照だけを書き出した文書に挿入します（変換は行われません）。

注：4th Dimension 2004 の新しいランゲージコマンドを使用して、XSL 変換を管理することもできます。詳細については、後述する「XSL 変換の管理」の節を参照してください。

ODBC データソースを使用した読み込みと書き出し

4th Dimension 2004 では、ODBC データソースからのデータ読み込みや、データソースへのデータ書き出しを行うことができます。この新しい機能により、4th Dimension と他のアプリケーションとのデータ交換の可能性が広がります。

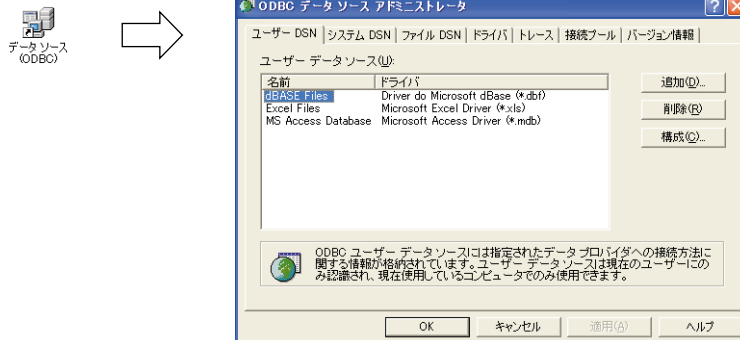
注：4th Dimension 2004 では、特定の新しいランゲージコマンドを使用して ODBC アプリケーションへ接続することもできます。詳細については、後述する「外部データソース」の節を参照してください。

ODBC データソースとは？

ODBCプロトコル（Open DataBase Connectivity）は、様々なアプリケーションがSQLランゲージを用いて互いに通信を行うためのインタフェースです。ODBCデータソースは、アプリケーションが開かれていない場合でも、そのアプリケーションに属すドキュメントに対して常に開かれた“窓”のようなものです。多くのアプリケーション（データベース、スプレッドシート、データマネージャなど）では、特定のODBCドライバを使用し、ODBCソースとしてデータを公開することができます。

ODBCプロトコルをサポートする任意のアプリケーションは、ODBCソースのデータの読み込みや書き出しを行えます。4th Dimensionの読み込み機能を使用すると、ODBCソースのデータを取得することができます。また、4th Dimensionの書き出し機能により、データをODBCソースへ挿入することができます。

■ Windowsにおいて、ODBCソースの定義は「管理用ツール」を用いて、「データソース (ODBC)」設定パネルで行います。



■ Mac OS Xにおいて、ODBCソースの定義は「ユーティリティ」フォルダの「ODBCアドミニストレータ」プログラムを使用して行います。



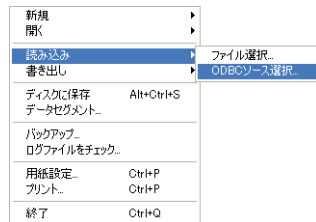
注：これらのダイアログボックスに関する詳細は、OSのドキュメントを参照してください。

ODBC ソースからのデータ読み込み

ODBC ソースからのデータ読み込みの概念は、他のタイプのデータ読み込みと同じです。ただし、ODBC ソースを選択する際に中間ダイアログボックスを使用する点が異なります。

▼ ODBC ソースからデータを読み込むには、次の手順に従ってください。

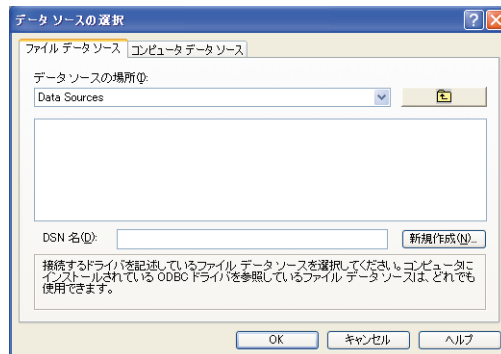
- 1 4th Dimension の「ファイル」メニューから「読み込み>ODBC ソース選択」コマンドを選択する。



ODBC ソースを選択するダイアログボックスが表示されます。

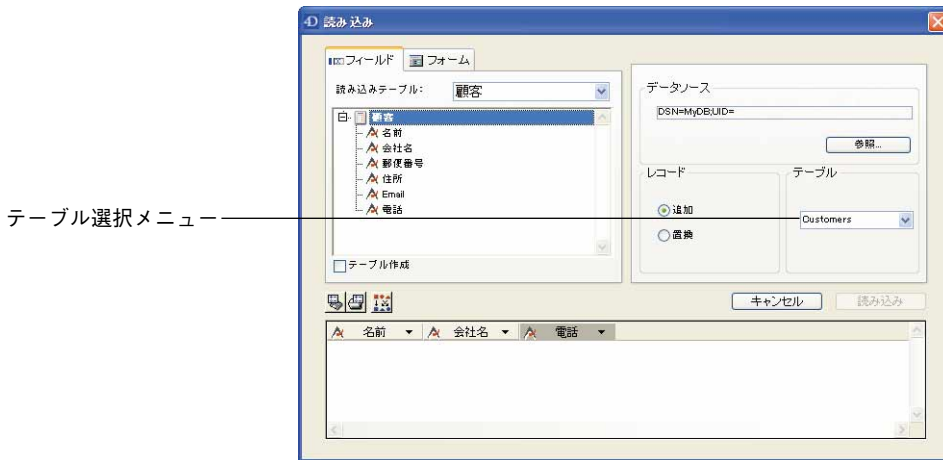
Windows

Mac OS



- 2 使用するデータソースを選択して「OK」(Windows) または「ログイン」(Mac OS) を選択する。

4th Dimension 標準のデータ読み込み用ダイアログボックスが表示されます。



- 3 ダイアログボックスの右側で、データの読み込み元である SQL テーブルを設定します (テーブル選択メニューを使用)。
- 4 データの読み込み先となるテーブルと4Dフィールドを選択するか、または「テーブル作成」オプションをチェックする (コンパイル済データベースに対してこのオプションは使用不可)。

他のすべてのタイプの読み込みと同様に、プレビューエリアには読み込みデータをフィールドに割り当てた状態が表示されます。この割り当ては変更することができません。

注: 「フォーム」タブをクリックし、フォームを用いてデータを読み込むこともできます。

- 5 ダイアログボックスの右側にある読み込みオプションを選択する。

■ データソース: このエリアには、データベースが接続されているデータソースの記述が表示されます。「参照...」ボタンをクリックすると、データソースを選択するダイアログボックスを再び表示することができます。

■ レコード: ソースから読み込むデータを読み込み先テーブルのレコードに追加したり、または置き換えることができます。

- 6 「読み込み」ボタンをクリックしてデータを読み込む。

データの読み込み後、4th Dimension とデータソースとの接続は自動的にクローズされます。

読み込み用のパラメータをすべてプロジェクトファイルに保存し、ダイアログボックスや **ODBC IMPORT** コマンドを使用して、後でそれをロードすることもできます。

これを行うには、ダイアログボックスの左下にあるプレファレンス保存用のボタンをクリックします。

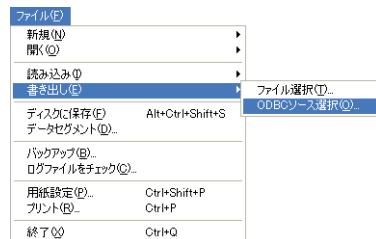
注：ODBC 読み込み用のダイアログボックスに保存したプロジェクトは、4th Dimension 標準の読み込み用ダイアログボックスと互換性がありません。

ODBC ソースへのデータ書き出し

ODBC ソースへのデータ書き出しの概念は、他のタイプのデータ書き出しと同じです。ただし、ODBC ソースを選択する際に中間ダイアログボックスを使用する点が異なります。また、書き出されたデータはディスクファイルには保存されず、公開された ODBC データソースへ直接保存されます。

▼ ODBC ソースへデータを書き出すには、次の手順に従ってください。

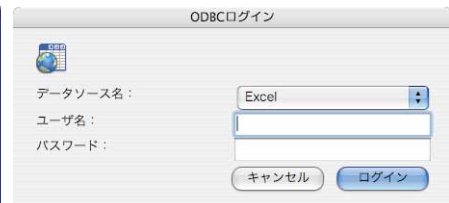
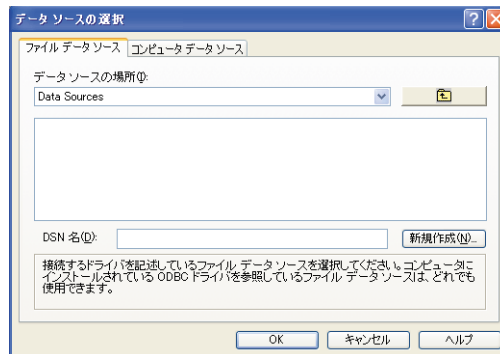
- 1 4th Dimension の「ファイル」メニューから「書き出し>ODBC ソース選択」コマンドを選択する。



ODBC ソースを選択するダイアログボックスが表示されます。

Windows

Mac OS



- 2 使用するデータソースを選択して「OK」(Windows) または「ログイン」(Mac OS) を選択する。

4th Dimension 標準のデータ書き出し用ダイアログボックスが表示されます。

テーブル選択メニュー



3 ダイアログボックスの右側で、データの書き出し先となるSQLテーブルを設定します (テーブル選択メニューを使用)。

4 書き出されるデータが属するテーブルと4Dフィールドを選択する。

他のすべてのタイプの書き出しと同様に、プレビューエリアには書き出しデータをフィールドに割り当てた状態が表示されます。この割り当ては変更することができます。

注：「フォーム」タブをクリックし、フォームを使用してデータを書き出すこともできます。

5 ダイアログボックスの右側にある書き出しオプションを選択する。

■ データソース：このエリアには、データベースが接続されているデータソースの記述が表示されます。「参照...」ボタンをクリックすると、データソースを選択するダイアログボックスを再び表示することができます。

■ レコード：書き出しを行えるのは、カレントセレクションまたはそのテーブルの全レコードだけです。

6 「書き出し...」ボタンをクリックしてODBCソースヘータを書き出す。

データの書き出し後、4th Dimensionとデータソースとの接続は自動的にクローズされます。

書き出し用のパラメータをすべてプロジェクトファイルに保存し、ダイアログボックスや**ODBC EXPORT**コマンドを使用して、後でそれをロードすることもできます。

これを行うには、ダイアログボックスの左下にあるプレファレンス保存用のボタンをクリックします。

注：ODBC書き出し用のダイアログボックスに保存したプロジェクトは、4th Dimension標準の書き出し用ダイアログボックスと互換性がありません。

クイックレポートエディタ

4th Dimension 2004の「クイックレポート」エディタには、レポートに対応する4th Dimensionコードの生成機能、ならびに新しい「標準偏差」演算子という2つの機能が新たに追加されています。

コードの生成

「クイックレポート」エディタに追加された新しいボタンを使用して、定義されたレポートに対応する4th Dimensionコードを自動生成し、それをクリップボードにペーストすることができます。この機能により、4th Dimensionランゲージでカスタマイズできるクイックレポートのアウトラインを手動で作成することができます。

この機能は「ユーザ」モードでのみ使用可能です。

コード生成を実行するための新しいボタンは、「クイックレポートウィザード」の最後のページにあります。



4Dコード作成

注：手動でクイックレポートを作成する場合でも、ウィザードの最終ページを表示して、この機能を利用することができます。

このボタンをクリックすると、ダイアログボックスが表示されます。このダイアログボックスを使用して、コード生成に関するパラメータを設定することができます。



次のオプションを利用できます。

- **境界線**：このオプションを選択し、生成された4Dコードに含めたいセル境界線のプロパティを指定します。
- **バレット表示**：このオプションを選択し、生成された4Dコードに含めたいカレントバレットの表示ステータス（標準、スタイル、カラムなど）を指定します。
オフスクリーンエリアでの使用を目的とするコードの場合、このオプションの選択を解除することをお勧めします。
- **テキストプロパティ**：このオプションを使用し、4Dコードに含めたいレポートのフォントスタイル定義（フォント、スタイルなど）を指定します。このオプションを選択しない場合、他の3つのオプションは無効になります。
 - **フォント、サイズ、整列**：このオプションを選択すると、クイックレポートテンプレートのフォントやフォントサイズ、整列に関する情報が生成された4Dコードに納められます。
 - **スタイル（太字、斜体、下線）**：このオプションを選択すると、クイックレポートテンプレートのテキストスタイルに関する情報が生成された4Dコードに納められます。
 - **カラー（テキスト、背景色）**：このオプションを選択すると、クイックレポートテンプレートのテキストカラーに関する情報が生成された4Dコードに納められます。

オプションの定義が終了したら、「コード作成」ボタンをクリックして4Dコードを生成します。すると、ダイアログボックスのプレビューエリアにそのコードが表示されます。



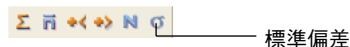
必要があれば、コード用のオプションを変更し、「コード作成」ボタンを再度クリックすることができます。変更に応じてプレビューエリアも更新されます。

満足できるコードが生成されたら、「クリップボードへコピー」ボタンをクリックし、次に「閉じる」ボタンをクリックしてダイアログボックスをクローズします。

この後、標準のショートカットキーである Ctrl キー+V (Windows) や Command キー+V (Mac OS)、または「編集」メニューの「ペースト」コマンドを使用して、メソッドやテキストファイルへコードをペーストすることができます。

標準偏差

「クイックレポート」エディタの「セル」メニュー、ならびに演算子パレットにおいて、新しい演算子である「標準偏差」を利用できるようになりました。



一連の値の標準偏差とは、平方偏差の平方根です。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「On a Series」の節を参照してください。

プログラムでこの演算子を使用できるように、2つの定数が新たに追加されています。

■ **QR EXECUTE COMMAND** コマンドと **QR Get command status** 関数に対し、「QR Commands」テーマ内に `qr cmd standard deviation(513)` が追加されました。

■ **QR GET TOTALS DATA** と **QR SET TOTALS DATA** コマンドに対し、「QR Operators」テーマ内に `qr standard deviation(32)` が追加されました。

QR SET TOTALS DATA コマンドの引数<値>にテキストや演算子を渡したい場合、このコマンドで「##D」コードを使用することができます。

4th Dimension 2004では、機能およびユーザインタフェースに関して数々の変更が加えられ、開発者の作業環境は一層向上しています。

- ユーザはログインダイアログボックスを用いて、パスワードの変更やアクセスの保存を行うことができます。
- 「環境設定」ダイアログボックスに新しいパラメータが追加されました。
- 各種エディタは、「ツールボックス」メニュー項目として新たに再編成されました（ユーザ、ユーザフォルダ、メニュー、ピクチャライブラリ、ヘルプメッセージ、リスト、スタイルシート、フィルタ、フォーマット）。
- 「デザイン」モードのユーザインタフェースは、さらに先進的な外観になりました（新しいツールバー、「デザイン」エディタの新しいプロパティ「インスペクタ」、エクスプローラの新しい外観）。
- エクスプローラにおけるオブジェクトフォルダ管理
- 「メソッド」エディタの新しい機能
- フォームエディタに関する変更：新しいインタフェース、新しいツール（ビュー、ズーム、整列ガイドなど）、新しいプロパティリストなど
- フォームオブジェクトに関する変更：新しい“リストボックス”オブジェクト、セパレータ、インタフェース、ネイティブデータ入力制御、3ステートチェックボックス（3つの状態をトグルする）、3Dボタンなど
- 同期化されたレコードリスト管理
- ユーザは、修正可能なフォームを指定して、それをオブジェクトライブラリから利用できるように設定することができます。
- 新しいアプリケーションビルダ
- Webサービス：さらに進化したDOCサービス管理（サブスクリプトと公開）

ログインダイアログボックス

4th Dimension 2004では、標準の「ユーザの識別（パスワード）」ログインダイアログボックスが変更されています。

- 新しいボタンを使用して、ユーザはパスワードをオンザフライで変更することができます。
- 新しいボタンを使用して、ユーザはデータベースアクセスパラメータをファイルに保存することができます（4D Clientのみ）。

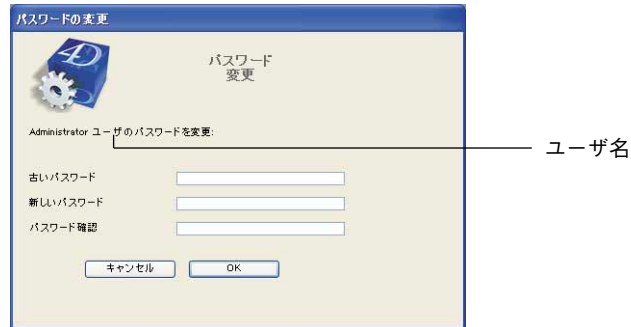
パスワードの変更

「ユーザの識別（パスワード）」ログインダイアログボックス（パスワード保護されたデータベースのオープン時に表示される）に「変更」ボタンが追加されています。



このボタンを使用し、カレントユーザのパスワードを変更するログインダイアログボックスにアクセスすることができます。ユーザリストが表示されている場合は、選択されているユーザがカレントユーザです。リストが表示されない場合、ログインダイアログボックスにカレントユーザ名を入力しなくてはなりません（ユーザリスト表示の設定は、データベースの「環境設定」で行います）。

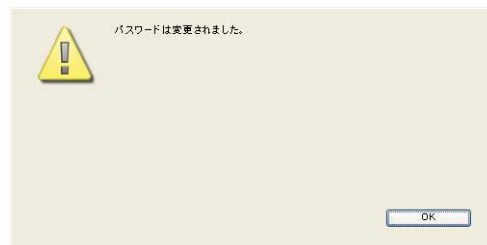
「変更」ボタンをクリックすると、次のログインダイアログボックスが表示されます。



パスワードを変更するには、まず初めに前のパスワードを入力しなくてはなりません。次に、新しいパスワードを入力し、それを確定します。

注：4th Dimension 2004では、カスタマイズしたダイアログボックスを使用して、ユーザパスワード制御システムを設定することができます。詳細については、後述の「独自のアクセス管理ダイアログボックスを提供する」の節を参照してください。

ダイアログボックスを確定し、変更が受け入れられると、4th Dimension からパスワードが正常に変更されたことを通知するメッセージが表示されます。

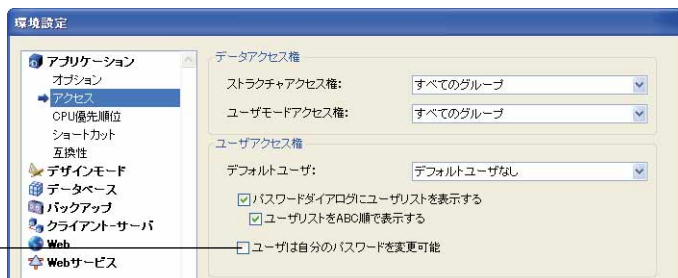


「デザイン」モードにおいて、データベースの以前のユーザパスワードが新しいもので置き換えられます。この後、ユーザは「接続」ボタンをクリックしてデータベースをオープンすることができます。

「変更」ボタンを表示しない

ユーザがパスワードを変更できないように、「変更」ボタンを隠すことができます。これを行うには、データベースの「環境設定」の「アプリケーション/アクセス管理」ページで「ユーザによるパスワード変更可」オプションの選択を解除してください。

「変更」ボタンの表示



互換性上の理由から、以前のバージョンの4th Dimensionで作成され、バージョン2004へ変換されたデータベースでは、デフォルトとしてこのボタンが表示されません。

データベースアクセスパラメータの保存

4D Client 2004では、「ユーザの識別」ダイアログボックスに「保存」ボタンが新たに追加されています。



アクセスパラメータ保存ボタン

注：パスワード保護機能が有効である場合は、4D Server データベースを選択した直後に、このダイアログボックスが表示されます。

このボタンを使用して、4D Server データベースのアクセス設定を4D Client接続用ファイルに保存することができます。

接続用ファイルを4D Clientアプリケーション上へドラッグ&ドロップすると（またはそのファイルをダブルクリック）、接続ダイアログボックスを使用せずに直接4D Serverデータベースへアクセスすることができます。また、この接続ファイルにはユーザパスワード（暗号化）を納められます。この場合、4D Serverデータベースへのアクセスは即座に実行されます。

注：以前のバージョンの4D Clientにおいて、この機能は4D Clientマシンのパスワードエディタを使用して利用することができました。

▼ データベースアクセスパラメータを保存するには、次の手順に従ってください。

1 ユーザログインダイアログボックスで、レコードを保存したいユーザ名を選択、または入力（現在のパラメータにより決定）した後、対応するパスワードを入力する。

2 「保存」をクリックする。

すると、次のようなダイアログボックスが表示されます。



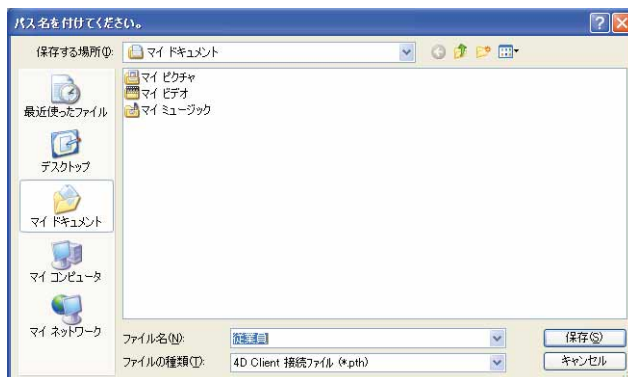
パスワードと一緒に、またはパスワードなしでアクセスパスを保存することができます。

■ **パスワード付き**：データベースへのアクセスパスとユーザパスワード（暗号化済）がドキュメントに保存されます。4D Clientの起動時にダイアログボックスは表示されず、即座に接続されます。

■ **パスワードなし**：データベースへのアクセスパスは保存されますが、データベースのオープン時にユーザのパスワードを入力しなくてはなりません。

3 「パスワード付き」または「パスワードなし」をクリックする。

標準のファイル保存ダイアログボックスが表示され、ここでハードディスク上のアクセスパスドキュメント名と保存場所を指定することができます。接続ファイルの拡張子は“.pth”です。



- 4 ファイル名と保存場所を選択し、保存用ダイアログボックスを確定する。
4D Client接続ファイルがディスクに保存されます。

環境設定






4th Dimension 2004の新機能（バックアップ、互換など）ならびに4D Customizer Plusユーティリティの統合によるオプションを盛り込むため「環境設定」ダイアログボックスは再編成され、その機能が強化されています。

4th Dimension 2004プロダクトラインでは、4D Customizer Plusが提供されません。このユーティリティを用いて今まで利用されていたオプションは、4th Dimensionの「環境設定」に統合されています。また、既に使われなくなったり不要となったオプションは削除されています。

比較表

次の表は、4th Dimension 2004の「環境設定」の設定と以前の設定場所（4th Dimension 2003の「環境設定」の各ページ、4D Customizer Plusまたは新機能）について説明したものです。

4th Dimension 2004の環境設定			以前の状況
テーマ	ページ	環境設定	
アプリケーション 	オプション	オプション	4D 2003デザインモード/オプション
		テンポラリフォルダの場所	4D 2003アプリケーション/システムリソース
		ドラッグ&ドロップ点減	4D 2003インタフェース/アピアランス
		進捗インジケータ	4D 2003インタフェース/アピアランス
		ツールバー表示 ¹	4D 2003インタフェース/アピアランス
		ウインドウ表示	4D Customizer Plus、後述の「ウインドウ表示」の節を参照
	アクセス管理	データアクセス権	4D 2003データベース/アクセス管理
		ユーザアクセス	4D 2003データベース/アクセス管理 + 4D 2004の新機能、前述の「変更ボタンを表示しない」の節を参照
	CPUの優先度	CPUの優先度	4D 2003アプリケーション/システムリソース
	ショートカット	キー	4D Customizer Plus、後述の「ショートカット」の節を参照
互換	ストラクチャの互換性	4D2003デザインモード/オプション +4D 2003インタフェース/アピアランス +4D 2004の新機能、後述の「新しい互換性オプション」の節を参照	
	Web互換性	4D 2003 Web/設定	
	プラットフォーム ²	4D 2003インタフェース/アピアランス	
デザインモード 	ストラクチャ	通常フォント	4D 2003デザインモード/フォント
		フォーム/メソッド自動コメント	4D 2003デザインモード/コメント
		ストラクチャエディタ	4D 2003デザインモード/ストラクチャエディタ
	フォームエディタ	オブジェクトテンプレート	4D 2004の新機能、後述の「フォームエディタ」の節を参照
		移動	
		自動整列	
		デフォルト表示	
	メソッドエディタ	マーカー	4D 2003デザインモード/メソッドエディタ
		フォント	
		デフォルト表示	
		オプション	
	コンパイラ	ストラクチャスタイルシート	4D 2003デザインモード/メソッドエディタ
		コンパイルオプション	4D 2003コンパイル/設定
	ドキュメント	コンパイラメソッド...	4D 2003コンパイル/設定
エクスプローラからのドキュメントアクセス		4D 2003デザインモード/ドキュメント	

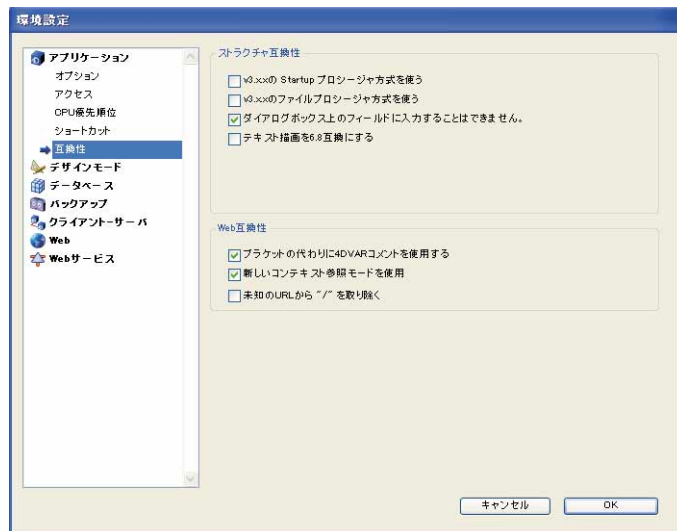
4th Dimension 2004の環境設定			以前の状況
テーマ	ページ	環境設定	
データベース 	データ管理	一般設定	4D 2003データベース/アクセス管理 + 4D 2003アプリケーション/システムリソース
		データベースキャッシュ設定	4D 2003アプリケーション/システムリソース + 4D 2004の新機能、後述の「メモリキャッシュの管理」の節を参照
		WEDD	4D Customizer Plus、後述の「WEDD」の節を参照
	スクリプトマネージャ	スクリプトマネージャ	4D Customizer Plus、後述の「スクリプトマネージャ」の節を参照
バックアップ 	設定		4D 2004の新機能、後述の「バックアップモジュールの統合」の章を参照
	スケジューラ		
	バックアップ		
	復元		
クライアント/サーバ 	設定	ネットワーク	4D Customizer Plus、後述の「ポート番号」の節を参照
		クライアント/サーバ接続タイムアウト	4D 2003アプリケーション/クライアント/サーバ
		クライアント/サーバ通信	4D 2003アプリケーション/クライアント/サーバ + 4D 2004の新機能、後述の「自動クライアント再接続機能を使用する」の節を参照
		4D Open	4D 2003アプリケーション/クライアント/サーバ
	公開	公開情報	4D Customizer Plus、後述の「公開」の節を参照
		許可-拒否テーブル設定	4D Customizer Plus、後述の「公開」の節を参照
		暗号化	4D 2003アプリケーション/クライアント/サーバ
Web 	設定	Webサーバ公開時オプション	4D 2003 Web/公開 + 4D 2004の新機能、後述の「HTTPSポート番号を変更する」の節を参照
		デフォルトHTMLパス	4D 2003 Web/公開
		開始時のモード	4D 2003 Web/公開
	詳細	キャッシュ	4D 2003 Web/設定
		Webプロセス	4D 2003 Web/設定
		オプション	4D 2003 Web/設定
		Webパスワード	4D 2003 Web/公開
	オプション	テキスト変換	4D 2003 Web/設定
		4D WebSTAR	4D 2003 Web/4D WebSTAR
		Keep-Alive接続	4D 2004の新機能、後述の「持続的接続」の節を参照
Webサービス 	SOAP	サーバ側設定	4D 2003 Web/Webサービス
		クライアント側設定	4D 2003 Web/Webサービス

1. 「カスタムメニュー」でのみツールバーの表示/非表示を設定できます。
2. 4th Dimension 2004では、プラットフォームインタフェース管理が変更されています。後述の「プラットフォームインタフェース」の節を参照してください。

主要な新機能については、後述されています。

新しい互換性オプション

「アプリケーション」テーマの「互換」ページに新しい互換性オプションが提供されています。



■「ダイアログボックスでフィールドを入力不可にする」

以前のバージョンの4th Dimensionでは、ダイアログボックス上でフィールドを用いて変数を入力することができませんでした（例えばDIALOGコマンドで表示されたダイアログ）。4th Dimension 2004では、この制約がなくなります。

特にデータベースでフィールドを使用し、データをダイアログボックスに表示している場合は、以前の動作をそのまま維持することができます。

デフォルトでは、バージョン2004に変換されたデータベースに対しこのオプションが選択されており、バージョン2004で作成されたデータベースに対しては選択されていません。

■ラジオボタンを名前でグループ化

このオプションにより、従来のラジオボタンに対して行われていた、頭文字に基づくグループ分け方法を維持することができます。詳細については、後述の「新しいラジオボタンの動作」の節を参照してください。

このオプションは、バージョン2004に変換されたデータベースにのみ表示されます。デフォルトでは、このオプションが選択されています。

バージョン2004で作成されたデータベースでは、新しいモードを使用してください。

■ PRINT SELECTION中にレコード毎にフォームをリロード

以前のバージョンの4th Dimensionでは、**PRINT SELECTION** コマンドによる印刷の際に使用されるフォームは、レコードごとにリロードされていました。これにより、開発者がランゲージを使用して「On printing detail」フォームイベント内で変更した可能性のあるすべてのオブジェクト設定が、自動的に再初期化されていました。

パフォーマンスの最適化を図るため、4th Dimension 2004ではこのメカニズムが削除されました。今後、4D開発者は独自に目的の設定をフォームメソッドで再初期化しなければなりません。これは、リストフォームが「On display detail」フォームイベントで行う動作と同じです。

しかし、「**PRINT SELECTION**中にレコード毎にフォームをリロード」オプションを使用すると、以前の動作を維持することができます。

このオプションは、バージョン2004に変換したデータベースにのみ表示されます。デフォルトでは、このオプションが選択されています。

バージョン2004で作成したデータベースでは、新しいモードを使用してください。

■ 未知のURLの“/”を取り除く

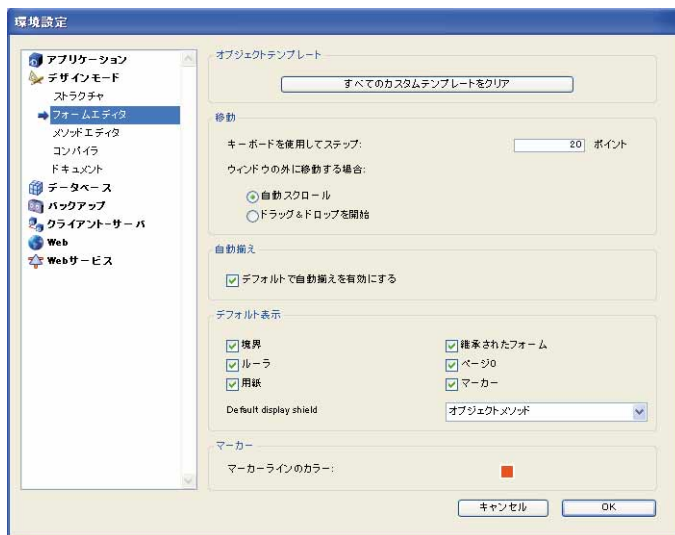
以前のバージョンの4th Dimensionにおいて、未知のURL（既存のページにも4D特定のURLにも該当しないURL）は、「On Web Authentication」および「On Web Web Connection」データベースメソッドに返され、“/”記号で開始しませんでした。4th Dimension 2004では、この特例が削除されています（後述の「未知のURLのサポート」の節を参照）。

しかし、この特例に基づいたアルゴリズムを採用しており、以前の動作を維持したい場合は、「未知のURLの“/”を取り除く」オプションをクリックしてください。

デフォルトでは、変換後のデータベースに対してこのオプションが選択されており、新しいデータベースでは選択されていません。

フォームエディタ

「フォーム」エディタページ（「デザインモード」テーマ）に新しいオプションが追加され、デフォルトのエディタ機能を指定できるようになりました。



■ カスタムテンプレートのクリア

このボタンを使用し、すべてのオブジェクトテンプレートをデフォルト値に再初期化することができます。詳細については、後述の「テンプレートとして使用」の節を参照してください。

■ キーボードによる移動単位

このエリアには、キーボードやShiftキーを用いてオブジェクトの移動やサイズ変更を行う際に使用する単位となる値を（ピクセル単位で）指定できます。

■ ウィンドウ領域を超えた移動時

このオプションを使用し、ウィンドウ領域を超えてマウスでオブジェクトを移動した場合の「フォーム」エディタの動作を指定できます。

■ **自動スクロール**：このオプションを選択すると、スクロールバーをクリックした場合と同様に、この動作によってウィンドウ上のフォームがスクロールされます。サイズの大きなフォーム上でオブジェクトを移動する場合、この動作が役立ちます。

■ **ドラッグ&ドロップ開始**：このオプションを選択すると、その動作がドラッグ&ドロップであるものと判断されます。フォームウィンドウは変更されず、移動したオブジェクトは別ウィンドウ（例えば、別のフォーム）にドロップされます（オブジェクト内容に互換性がある場合）。複数フォーム間でオブジェクトを再利用する場合、この動作が役立ちます。

作業パターンや開発上のニーズに応じて、このオプションを設定することができます。デフォルトでは、「自動スクロール」オプション（従来の4th Dimensionの動作）が選択されます。

■ 自動整列

このオプションを使用すると、「フォーム」エディタの各新規ウインドウにおける自動整列をデフォルトとしてアクティブにすることができます。各ウインドウごと個別に、このオプションを変更することが可能です（後述の「マグネティックグリッドを使用する」を参照）。

■ デフォルト表示

このオプションを使用し、「フォーム」エディタの各新規ウインドウにおける項目の表示／非表示をデフォルトとして指定したり、マーカーラインのカラーやデフォルトの表示バッジを設定することができます。

「フォーム」エディタの「表示」階層メニューを使うと、表示／非表示を各項目ごとに指定できます。

バッジに関する詳細は、後述の「バッジを使用する」の節を参照してください。

メモリキャッシュの管理

メインメモリについて：4th Dimension 2004では、Windows XPやMac OS Xの標準に合わせて、メインメモリの管理が最適化されています。今後、4th Dimensionが使用するメモリはシステムにより自動的に割り当てられ、カスタマイズできなくなります。ただし、使用するデータベースのメモリキャッシュはカスタマイズすることができます。

4th Dimension 2004では、メモリキャッシュの設定が変更されています。キャッシュ計算モードは次の2種類です。

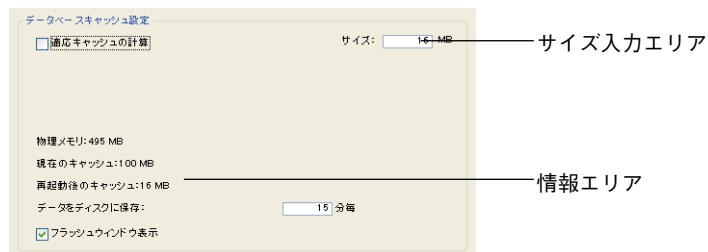
■ “強制” モード

■ 適応モード

強制モード

このモードでは、ユーザがデータベースのメモリキャッシュサイズを指定します。これを行うには、「データ管理」ページ（「データベース」テーマ）の「アダプティブ（適応）キャッシュの計算」オプションの選択を解除します。

すると、4th Dimensionにより入力エリアが表示され、使用するメモリキャッシュを設定できるようになります。さらに、物理メモリ（マシン上で使用可能なRAM）や、現在のキャッシュ、再起動後のキャッシュ（行った変更を反映）に関する情報が表示されます。

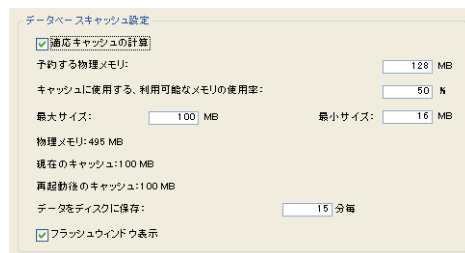


マシン資源の状況とは関係なく、入力するメモリキャッシュサイズが確保されます。

この設定は、一定の動作環境下、つまりメモリに関して異なるシステム上で利用できるように、データベースが設計されている場合に使用されます。大抵の場合、適応キャッシュの方がより高いパフォーマンスを得られます。

適応モード

このモードでは、設定された制限を考慮しながら、システムがメモリキャッシュの管理を動的に行います。これにより、ほとんどの動作環境に適応し、高いパフォーマンスが得られるメモリキャッシュが動作環境に割り当てられます。このモードに設定するには、「アダプティブキャッシュの計算」オプションをチェックします。すると、4th Dimension は複数の入力エリアを追加表示します。



これにより、設定したパラメータに応じてメモリキャッシュサイズが動的に計算されます。デフォルトとして提示される値は、4th Dimension の標準的な使用に対応しています。

■ **予約する物理メモリ**：システムおよび他のアプリケーション用に確保するRAMメモリの割り当て。4th Dimension と同じマシン上で他のアプリケーションが動作している場合、最適化のためにこの値が増やされます。

■ **空きメモリ中キャッシュに使用する率**：デフォルトとしてキャッシュに割り当てる残りのメモリの割合。

デフォルトとしてキャッシュに割り当てられたサイズを取得するには、次の計算を実行します。

(物理メモリ - 予約する物理メモリ) × キャッシュに使用されたメモリの割合

適応モードにおけるメモリキャッシュのサイズは、アプリケーションとシステムの需要に応じて動的に変化します。次の2つのオプションを使用して、制限を設定することができます。

■ **最大サイズ**：動的にキャッシュに割り当てられるメモリ量の最大サイズ。最大サイズは2,000MB (2GB) を超えてはいけません。

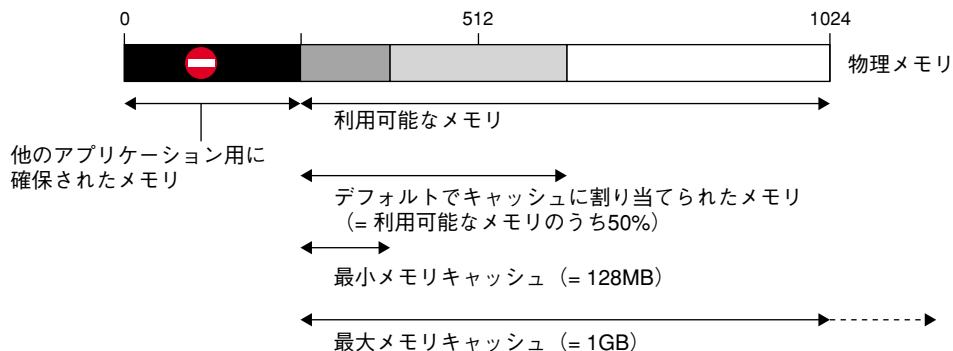
■ **最小サイズ**：動的にキャッシュに割り当てられるメモリ量の最小サイズ。最小サイズは4MB より小さくすることはできません。

注：各マシン上に配布されるデータベースの場合、事前にメモリ構成がわからないため、この制限を設定しておくると特に役立ちます。その場合、設定した制限により、あらゆる状況において最低限のパフォーマンスが保証されます。

次の図は、この動作を示したものです。

メモリキャッシュの計算例

割り当てる物理メモリ = 256 MB
空きメモリ中キャッシュに使用する率 = 50%
最大サイズ = 1 GB 最小サイズ = 128 MB



動的モードでの設定例

次の例は、さまざまなソフトウェアおよびメモリ設定において、動的キャッシュを使用する利点を示しています。

■ 768MB の RAM を搭載したコンピュータで1つの4D Serverが動作する場合。

この場合、最大限のメモリキャッシュを割り当てるとよいでしょう。

■ 予約する物理メモリ = 256MB

- 利用可能なメモリからキャッシュに使用する率=100% (4D Serverアプリケーションが1つしか動作していないため)

- 最小サイズ=10MB、最大サイズ=1GB (この設定において、これらの値は役に立ちません)

デフォルトキャッシュサイズ： $(768-256) \times 100\% = 512\text{MB}$

- 前述の例と同じ設定で、そのコンピュータ上で3つの4D Serverが動作する場合。それぞれのアプリケーションに対して同量のキャッシュを割り当てたいものとします。

この場合、各サーバに対して「利用可能なメモリからキャッシュに使用する率」を33%に設定する必要があります。

各アプリケーションのデフォルトキャッシュサイズ： $(768-256) \times 33\% = 170\text{MB}$

- 数種類の設定を使用して、広範に配布するための実行形式アプリケーションを作成する場合。検証の結果、20MBから100MBまでのキャッシュを割り当てると、アプリケーションが最善の状態で作動することがわかったものとします。

- 予約する物理メモリ=256MB

- 利用可能なメモリからキャッシュに使用する率=50% (他のアプリケーションから利用できるメモリを残しておくため)

- 最小サイズ=20MB、最大サイズ=100MB

256MBのRAMを搭載したマシンの場合、4th Dimensionは最小キャッシュサイズ=20MBを使用します。

1GBのRAMを搭載したマシンの場合、4th Dimensionは最大キャッシュサイズ=100MBを使用します。

384MBのRAMを搭載したマシンの場合、4th Dimensionは、 $(384-256) \times 50\% = 64\text{MB}$ を使用します。

自動クライアント再接続機能を使用する

「クライアント/サーバ」テーマの「設定」ページにおいて、新しいオプションである「自動クライアント再接続機能を使用する」(バージョン2003.3以降の4th Dimensionより追加)を利用することができます。このローレベル機能により、ある特定の動作環境において、予期しない接続切断の際にクライアントマシンへ自動的に再接続することができます。

このオプションは、すべてのクライアントマシンを対象とします。このオプションを選択した場合、クライアントとの接続が失われると、即座にクライアントへの自動再接続機能が使用されます。再接続は一度だけ行われます。再接続が失敗に終わった場合、エラー-10002が返されます。正常に接続された場合、ウィンドウのタスクバーに通知アイコンが表示されます。

注：クライアントへの自動再接続機能を利用可能にするには、サーバ側のタイムアウトの値を1分以上に設定しなくてはなりません。

4D Customizer Plus から統合された新機能

4th Dimension 2004 プロダクトラインでは、4D Customizer Plusユーティリティが提供されません。その主要な機能は、4th Dimensionの「環境設定」に統合されています（後述）。

ウィンドウを表示する

アプリケーションの「環境設定」の「アプリケーション/オプション」ページにある「ウィンドウを表示する」エリアを使用し、アプリケーションで2つのタイプのウィンドウをデフォルトとして表示するかどうかを指定することができます。

- **プリントの進捗状況を表示する**：このオプションを使い、印刷中のプリント進捗ダイアログボックスの表示を有効、または無効にすることができます。
- **スプラッシュスクリーンを表示する**：このオプションを無効にすると、「ユーザ」または「カスタム」モードにおいて、スプラッシュスクリーンが表示されません。



このウィンドウを隠した場合、すべてのウィンドウの表示はプログラムから独自に処理してください。

注：4D Customizer Plusにおいて、このオプションは「ウィンドウ」タイプのポップアップメニューから「Hide window」を使って利用することができました。

ショートカット

「環境設定」の「アプリケーション」テーマにあるこのページを使用して、4th Dimensionの3つの基本的な操作（入力フォーム受け入れ、入力フォームキャンセル、サブフォーム追加）に対するデフォルトのショートカットを表示したり、変更することができます。これらのショートカットは、双方のプラットフォームで共通です。キーの形のアイコンは、対応するWindowsまたはMac OSのキーを示します。デフォルトのショートカットは次の通りです。

■ 入力フォーム受け入れ：Enter

■ 入力フォームキャンセル：Esc

■ サブフォーム追加：Ctrl+Shift+ “/”（Windows）またはCommand+Shift+ “/”（Mac OS）

任意の操作のショートカットを変更するには、対応する「編集」ボタンをクリックします。すると次のようなダイアログボックスが表示されます。



ショートカットを変更するには、キーボード上で新しいキーコンビネーションをタイプし、「OK」をクリックします。

ある処理に対してショートカットを設定したくない場合、「クリア」ボタンをクリックします。

WEDD

「データベース」テーマの「データ管理」ページにWEDDエリアが設けられ、WEDD署名を指定できるようになりました。

WEDD署名は、ストラクチャファイルをデータファイルに関連付けます。WEDD署名を含むストラクチャファイルは、同じWEDD署名を含むデータファイルでのみ開くことができます（またはその逆）。これにより、互換性のないデータファイルの使用を回避することができます。複数のデータベースを用いて作業をしており、特定のストラクチャに属さないデータファイルを誤って開かないようにしたい場合にも、WEDD署名を指定することができます。

デフォルトでは、データベースにこのパラメータは設定されていません。データベースにWEDD署名を追加するには、WEDDエリアに独自の値を入力します。この値として、任意のタイプの文字を組み合わせることができます。これにより、ストラクチャファイルと現在のデータファイルが関連付けられます。

WEDD署名を削除するには、このエリアの内容を削除します。

スクリプトマネージャ

「データベース」テーマの「スクリプトマネージャ」ページにおいて、スクリプトマネージャをともなうデータベースを使用する場合に、特定のパラメータを変更することができます。スクリプトマネージャは、ローマ字以外の言語（日本語、アラビア語など）の記述を処理するシステムの一部です。

4th Dimension をスクリプトマネージャとともに使用する場合、このウインドウ上のいくつかのパラメータを変更する必要があるかもしれません。

- **0（ゼロ）ASCIIコード**：ゼロを表す文字のASCIIコードです。日本語システムのデフォルト値は48です。他のシステムではこの値が異なる場合があります。
- **ゼロの次の文字**：ゼロの次のバイトの値です。このパラメータは、アラビア語またはヘブライ語のシステム用にローカライズする場合にのみ使用します。その他の場合、このパラメータを0（ゼロ）に設定してください。
- **比較モード**：このドロップダウンメニューには、並び替えや検索に関する4つの選択項目が用意されています。
 - **4th Dimension**：TRICリソース、またはデフォルトとしてTRI#リソースを使用する。
 - **System（システム）**：ROMの比較用ルーチンと大文字用ルーチンを使用する（TRICリソースは無視する）。
 - **German for V2.2（ドイツV2.2用）**：“ β ”文字の処理を伴う、ドイツで使用される比較方法。
 - **Turkish（トルコ）**：トルコで使用される比較方法。
- **TRICリソース（新規データベース）とTRICリソース（オープンデータベース）**：デフォルトでは、4th Dimensionから提供されるテーブルに基づいて並び替えが行われず（“e”、“é”、“è”、“ê”、“ë”は別々に並び替えられますが、“é”を検索すると“e”を検出します）。このテーブルは、言語によって異なります（特殊文字など）。システムの言語に応じてTRICリソースを変更することができます。2種類の使用方法に対して、2つのメニューが用意されています。

- 「新規データベース」メニューを使用し、4th Dimensionで作成されたすべてのデータベース用のTRICリソースを変更することができます。単一システムで利用するデータベースを開発する場合、このメニューを使います。
- 「オープンデータベース」メニューを使用し、カレントデータベースだけに対応するTRICリソースを変更することができます。特定のシステムで使用するデータベースを特別に開発する場合、このメニューを使います。

注：TRICリソースを変更すると、4Dのソートテーブルは再構築され、データベースのインデックスが自動的に再作成されます。

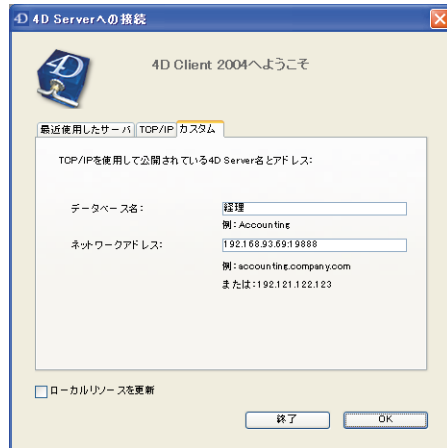
- 日付計算：このオプションを使用して日付計算用の4th Dimension設定を調整します。ペルシャ暦の場合、その計算には4バイトを使用します。その他の場合は、8バイトが必要です。

ポート番号

「クライアント／サーバ」テーマの「設定」ページにあるこのオプションを使用して、4D Serverがデータベースを公開するTCPポート番号を変更することができます。この情報は、データベースストラクチャファイルと各クライアントマシン上に保存されます。デフォルトでは、4D Serverと4D Clientが使用するTCPポート番号は19813です。

TCPプロトコルを使用し、同じマシン上で複数の4Dアプリケーションを利用したい場合は、この値をカスタマイズする必要があります。この場合、各アプリケーションに対して別々の値を設定しなければなりません。

4D Serverや4D Clientからこの値を変更すると、その値はデータベースに接続するすべての4D Clientマシンへ自動的に送信されます。接続していない他のクライアントマシンを更新するには、そのクライアントの次回接続時に、「接続」ダイアログボックスの「カスタム」ページにおいて、サーバマシンのIPアドレスに続けて新しいポート番号（先頭にコロンを付加）を入力してください。例えば、新しいポート番号が19888である場合は、次のように指定します。



注：4D Clientで定義したポート番号と同じ番号を使って公開されたデータベースだけが、TCP/IP ダイナミック公開ページに表示されます。

公開

「クライアント／サーバ」テーマの「公開」ページには、4D Serverによるデータベース公開に関連する新しいオプションがいくつか追加されています。

- **公開情報：**このプロパティを使用し、ネットワーク上における4D Serverデータベースの公開モードを指定することができます。2種類のオプションを利用できます。
- **公開しない：**データベースは公開されず、公開されたデータベースの一覧に表示されません（TCP/IPページ）。接続を行うには、接続ダイアログボックスの「カスタム」ページにおいて、ユーザがデータベースのアドレスを手動で入力しなければなりません。
- **公開する：**データベースは公開され、公開されたデータベースリストの一覧に表示されます（デフォルトのオプション）。
- **許可-拒否テーブル設定：**このテーブルを使用し、4D ClientマシンのIPアドレスごとにデータベースへのアクセス制御ルールを設定することができます。例えば、戦略的アプリケーションを使用する場合は、このオプションによりセキュリティを強化することができます。

注：この設定テーブルにより、Web接続が制御されることはありません。

接続テーブルの機能を次に示します。

- 「許可-拒否」カラムには、ポップアップメニューを使用して、適用するルールのタイプ（許可または拒否）を選択することができます。ルールを追加するには、「追加」ボタンをクリックします。すると、テーブルに新しい行が表示されます。「削除」ボタンを使用すると、現在の行を削除することができます。
- 「IPアドレス」カラムには、そのルールに関するIPアドレスを設定することができます。アドレスを設定するには、そのカラムをクリックして“123.45.67.89”という形式でアドレスを入力します。

“*（アスタリスク）”記号を用いて、“～で開始”タイプのアドレスを指定することができます。例えば、“192.168.*”と指定すると、“192.168.”で始まる全アドレスを表わします。
- 各ルールは、テーブルの表示順に適用されます。2つのルールが相反する場合、テーブルの最上位に位置するルールが優先されます。

現在の並び順を変更して、行の順序を再編成することができます（カラムのヘッダをクリックし、並び替え方向を変更する）。また、ドラッグ&ドロップを使って行を移動することも可能です。
- セキュリティ上の理由から、実際にルールと一致するアドレスだけが接続を許可されます。言い換えれば、テーブルに1つ以上の「拒否」ルールだけが含まれている場合、ルールに一致するものが一つもないため、すべてのアドレスが拒否されます。一部のアドレスだけを拒否したい場合（その他のアドレスは許可する）、テーブルの最後に「許可*」ルールを追加してください。

例えば

- ・拒否 192.168.*（192.168で始まるアドレスをすべて拒否する）
- ・許可*（しかし、その他のアドレスはすべて許可する）

デフォルトでは、4D Serverによる接続制限は一切適用されません。テーブルの最初の行には、「許可」というラベルと“*”記号（すべてのアドレス）が納められています。

インタフェース

4th Dimensionの「デザイン」モードにおいて、大部分のエディタのインタフェースが変更されています。

ツールバー

新しいツールバー（「デザイン」モードおよび「ユーザ」モード）はさらに機能が向上し、主要な機能へのダイレクトアクセスを提供します。

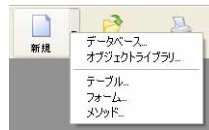
「デザイン」モード



「ユーザ」モード

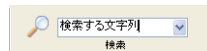


三角のアイコンが付いたボタンには、メニューが関連付けられています。



これらのボタンの動作は次の通りです。ボタンをクリックすると、現在関連付けられている動作を実行します。この動作は、前回メニューから選択されたものです。メニューを表示して別の動作を選択するには、ボタンの右側をクリックします。

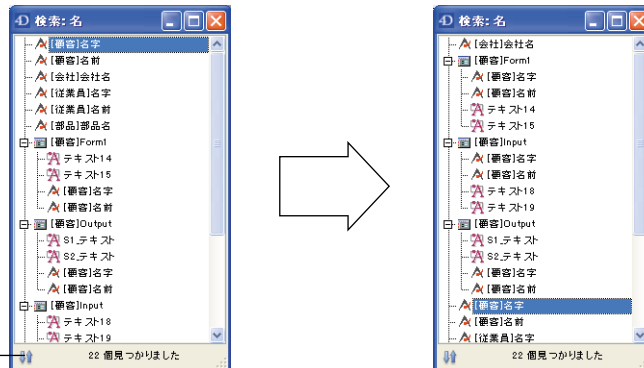
- 「検索」エリアを使用し、標準のデータベース検索を開始することができます。検索を直接開始するには、入力エリアに検索する文字列を入力してReturnキーを押します。



オプションなし（全データベースオブジェクトの“含む”タイプ）で検索が実行されます。

オプション付きで検索を実行したい場合、虫メガネの形のアイコンをクリックし、標準の検索ダイアログボックスを表示してください。

検索結果を示すウインドウには新しいアイコンが追加され、これを使用するとウインドウの内容（リストおよび項目）をアルファベット順に並び替えられる点に注目してください。



並び替えアイコン

メニューの再編成

4th Dimension のメニューバー（「ユーザ」モードおよび「デザイン」モード）が整理され、より標準的なロジックが提供されています。この再編成により、さらに直感的にアプリケーションの各種機能へアクセスすることができます。

この節では、メニューの主な変更点について説明します。

- 「デザイン」モードにおいて、「ファイル」メニューと「編集」メニューが変更され、「実行」メニューと「ウインドウ」メニューが追加されました。
- 「ユーザ」モードにおいて、「ファイル」メニューと「編集」メニューが変更され、「レコード」メニューと「ツール」メニューが追加されました。

関連するエディタのメニューの変更については、それぞれのエディタのドキュメントで説明しています。

「デザイン」モード

次の表は、4th Dimension 2004の「デザイン」モードにおける主な新規コマンドと変更されたコマンドの説明、ならびに前バージョンの4th Dimensionの対応するコマンドを示しています。

「デザイン」モードの変更されたコマンド	4th Dimension の対応するコマンド
ファイル	
新規>データベース	ファイル/データベースを開く
新規>オブジェクトライブラリ	-
新規>テーブル	ストラクチャ/新規テーブル (デザインエディタ)
新規>フォーム	デザイン/新規フォーム
新規>メソッド	デザイン/新規メソッド
開く>データベース	ファイル/データベースを開く
開く>オブジェクトライブラリ	-
開く>フォーム	デザイン/フォーム編集
開く>メソッド	デザイン/メソッド編集
編集	
検索>検索コマンド	編集/検索コマンド
実行（「デザイン」および「ユーザ」モードで同一メニュー）	
コンパイル済み/インタプリタモード	モード/コンパイルモードで実行 モード/インタプリタモードで実行

メソッド	「ユーザ」モード/特別/メソッド実行
ランタイムエクスプローラ	ツール/ランタイムエクスプローラ
Webサーバ開始	「ユーザ」モード/Web Server/
	Web sever開始
Webサーバテスト	-
デザイン	
エクスプローラ>エクスプローラページ	ツール/エクスプローラ
ツールボックス>ユーザ	ツール/パスワード+
	パスワードメニュー
ツールボックス>ユーザグループ	ツール/パスワード+
	パスワードメニュー
ツールボックス>メニュー	ツール/メニューバーエディタ
ツールボックス>ピクチャライブラリ	ツール/ピクチャライブラリ
ツールボックス>ヘルプメッセージ	オブジェクトプロパティパレット
ツールボックス>リスト	ツール/リストエディタ
ツールボックス>スタイルシート	編集/環境設定、スタイルシートページ
ツールボックス>フィルタ	編集/環境設定、フォーマットページ
ツールボックス>ツールボックス	-
データベースストラクチャ	ツール/データベースストラクチャ
Webサービスウィザード	ツール/Webサービスウィザード
シンタックスチェック	ツール/コンパイラ
アプリケーションビルド	ファイル/アプリケーションビルド
ウインドウ (編成ならびにオープンウインドウの一覧)	-

「ユーザ」モード

次の表は、4th Dimension 2004の「ユーザ」モードにおける主な新規コマンドと変更されたコマンドの説明、ならびに前バージョンの4th Dimensionの対応するコマンドを示しています。

「ユーザ」モードの変更されたコマンド	4th Dimensionの対応するコマンド
ファイル	
新規>データベース	ファイル/データベースを開く
新規>データファイル	ファイルを開くダイアログボックスの
「他のデータベース...」をクリック	
開く>データベース	ファイル/データベースを開く
開く>データファイル	ファイルを開くダイアログボックスの
「他のデータベース...」をクリック	
読み込み>ファイル選択	ファイル/データ読み込み

読み込み>ODBC ソース選択	-
書き出し>ファイル選択	ファイル/データ書き出し
書き出し>ODBC ソース選択	-
データをディスクに保存	Ctrl+w/ Command+w
データセグメント	「デザイン」モード/ツール/ストラクチャ、 次にストラクチャ/データセグメント
バックアップ	-
ログファイルをチェック	-
編集	
環境設定 (Windows)	「デザイン」モード/編集/環境設定
実行 (「デザイン」および「ユーザ」モードと同一メニュー)	
コンパイル済み/ インタプリタモード	モード/コンパイルモードで実行
	モード/インタプリタモードで実行
メソッド	「ユーザ」モード/特別/メソッド実行
ランタイムエクスプローラ	ツール/ランタイムエクスプローラ
Web サーバ開始	「ユーザ」モード/Web サーバ/ Web サーバ開始
Web サーバ終了	「ユーザ」モード/Web サーバ/ Web サーバ終了
Web サーバテスト	-
レコード	
新規リストレコード	更新/リスト更新 + 更新/新規レコード
新規レコード	更新/新規レコード
レコード修正	更新/レコード修正
すべて表示	クエリ/すべて表示
サブセットを表示	クエリ/サブセットを表示
クエリ> Query コマンド	クエリ/Order by コマンド
並び替え	クエリ/並び替え
テーブル/フォーム選択	ファイル/テーブル/フォーム選択 + ショートカットキー Ctrl (Command) +スペースバー
フォーミュラで更新	更新/フォーミュラで更新
リスト更新	更新/リスト更新
ツール	
クイックレポート	レポート/クイックレポート
ラベル	レポート/ラベル
チャート	レポート/チャート
4D Chart	プラグイン/4D Chart

OLE tools	プラグイン/OLE tools
その他プラグイン	プラグイン/その他プラグイン
ウインドウ（オープンウインドウの編成および一覧）	-

注：4th Dimension 2003の「ユーザ」モードにおける既存の各種機能は、差し替えられたり、あるいは移動されています。

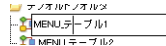
- ・ログファイル管理コマンド（「ファイル」メニュー）は、アプリケーションの「環境設定」の「バックアップ」テーマで取り扱われるようになりました。
- ・新しいリスト管理メカニズムにより不要となったため、「リスト更新」コマンド（「更新」メニュー）が取り除かれました（後述の「レコードリストの管理」の節を参照）。
- ・書き出しや読み込みフィルタ（「特別」メニュー）の編集コマンドは、アプリケーションの「環境設定」から利用できるようになります。

ショートカット

4th Dimension 2004では、ショートカットが再編成され、不要になったショートカットは削除されました。

- 「ユーザ」モードで「テーブルリスト」の表示に使用されていたショートカット、「Ctrl キー+スペースバー（Windows）」または「Command キー+スペースバー（Mac OS）」は、「レコード」メニューの「テーブルリスト」コマンドに置き換えられました。また、ショートカットは「Ctrl キー+0（Windows）」または「Command キー+0（Mac OS）」に変わります。
- 「デザイン」モードで「エクスプローラ」の表示に使用されていたショートカット「Ctrl キー+スペースバー（Windows）」または「Command キー+スペースバー（Mac OS）」は、ショートカット「Ctrl キー+”,”（Windows）」または「Command キー+”,”（Mac OS）」に置き換えられました。
- 「ユーザ」モードで強制的にキャッシュをディスク保存する際に使用されていたショートカット「Ctrl キー+W（Windows）」または「Command キー+W（Mac OS）」は、「ファイル」メニューの「データをディスクに保存」コマンドに置き換えられました。またショートカットは「Alt+Ctrl+Shift+S（Windows）」または「Option+Command+Shift+S（Mac OS）」に変わります。
- リスト項目（例えばエクスプローラ上のメソッド名やメニューバーエディタ上のメニューコマンド）に編集用カーソルを挿入するために使用されていたショートカット「Ctrl キー+クリック（Windows）」または「Command キー+クリック（Mac OS）」は、ショートカット「Alt キー+クリック（Windows）」または「Option キー+クリック（Mac OS）」に置き換えられました。リスト項目上でクリックを二回行うと、編集モードへ移行する点に注意してください。その場合、その項目全体が選択されます。

Altキー+クリック

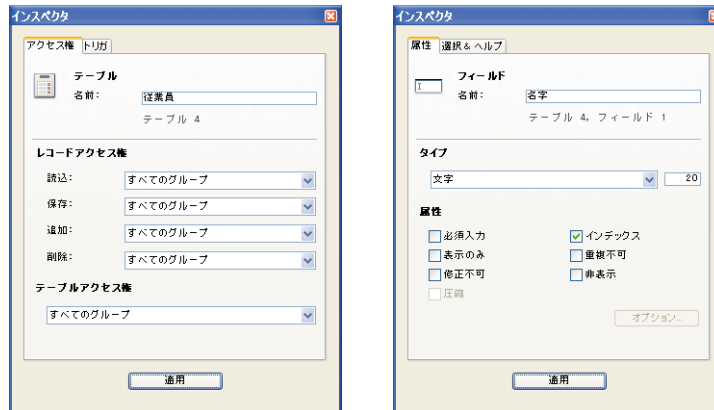


二回の単独クリック



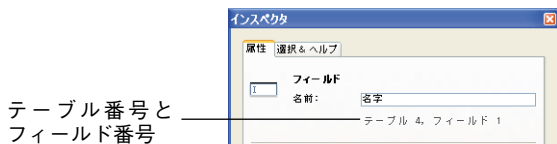
インスペクタ

「ストラクチャ」ウインドウ上でのテーブルやフィールド、リレートプロパティへのアクセスは、「インスペクタ」という単独のパレットを使用して行われるようになりました。



パレットの内容は動的であり、選択したオブジェクトに応じて更新されます。

パレットには、テーブル番号とフィールド番号が表示される点に留意してください。



テーブル番号と
フィールド番号

注：「インスペクタ」パレットを使用して、選択したオブジェクトにカラーを適用することはできません。このプロパティはコンテキストメニューでのみ処理できます。

ランタイムエクスプローラ

ランタイムエクスプローラの「ウォッチ」ページの内容が変更されました。

■より明確になるように、今まで「インフォメーション」リストに表示されていた項目は再編成され、「インフォメーション」および「Web」という2つのリストに分けられました。

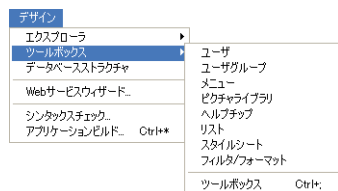
■ このページのコンテキストメニューの「モニタリングを有効にする」コマンドを選択すると、追加情報が提供されます。ただし、モニタリングを有効にすると処理速度が低下するため注意が必要です。

この追加情報は、「スケジューラ」や「ネットワーク」リストに対して表示され、これによりアプリケーションの内部的なアクティビティを監視する高度な機能を利用できます。

ツールボックス

4th Dimensionの補助的な各種エディタは、ツールボックスというひとつの環境にまとめられています。

「デザイン」メニューの「ツールボックス」階層コマンドを用いて、ツールボックスへアクセスすることができます。

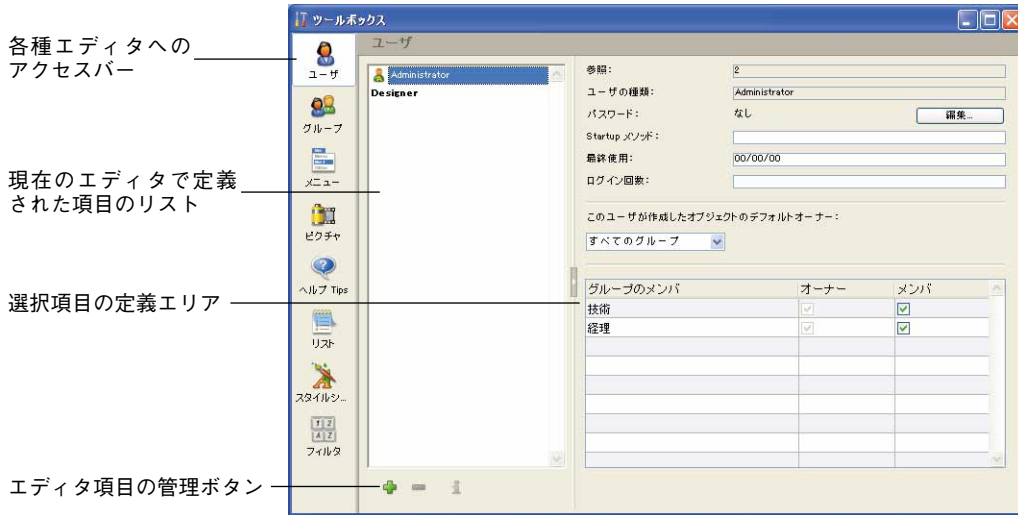


各サブコマンドは、対応するページ上にツールボックスを表示します。「ツールボックス」コマンドは、カレントページにツールボックスを開きます。

また、ツールバーの「ツール」ボタンを使用することもできます。



デフォルトとして、ツールボックスは最初のページを表示します（ユーザ）。このページには次の項目があります。



「ユーザー」と「グループ」エディタのアイコンは、そのシステムで使用されるアイコンに対応しています。これらのアイコンは、Mac OS X上で次のように表示されます。

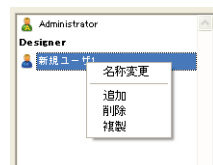


- リストの下側にあるボタンを使用すると、表示されるエディタに応じて項目の追加や削除を行うことができます（ユーザー、グループ、メニュー、ピクチャ、メッセージ、リスト、スタイルシート、フィルタ/フォーマット）。

項目を追加 ———   ——— 選択した項目を削除

一部のエディタにはさらにボタンが存在し、特定の機能を提供します。

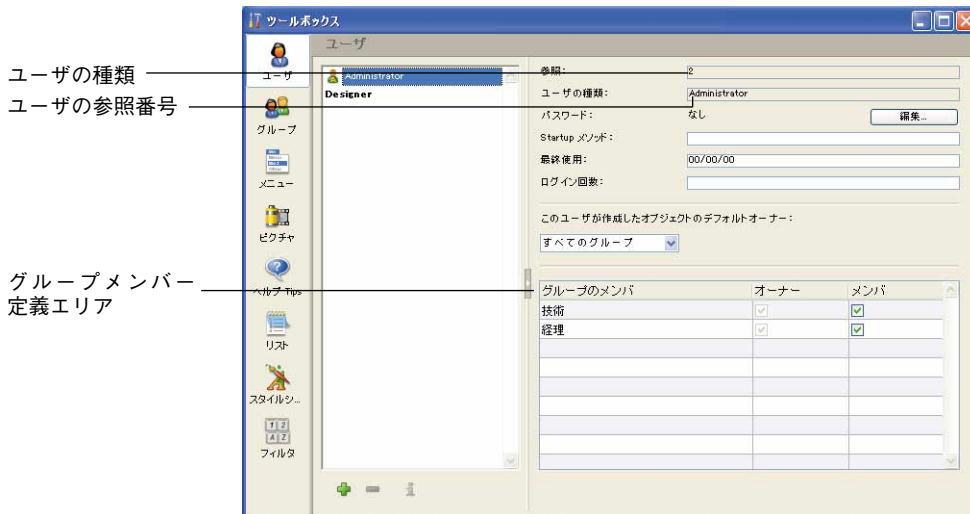
- また、項目リストにはコンテキストメニュー（Windowsでは右クリック、Mac OSではControl+クリックで表示）が付属し、項目の編集機能（追加、削除など）にアクセスすることができます。



ツールボックスエディタ内の機能は、概ね以前のバージョンの4th Dimensionの機能と同じですが、いくつかの例外があります（次のページで説明）。

ユーザ

このエディタを使用し、データベースユーザのプロフィールを定義することができます。



■ ユーザリストには、カレントユーザに対して“表示可能な”すべてのユーザが表示されます。

■ Designerの場合：すべてのユーザ

■ Administratorの場合：Administratorが作成したユーザのみ（緑のアイコン）

これら2つのユーザのアイコンには、それぞれ“D”と“A”という文字が表示されます。これら2つのユーザの名前は変更可能です。

このリストのコンテキストメニューを使用して、ユーザ名の変更、ユーザの追加、削除、コピーを行うことができます。



注：ユーザ名を変更するには、ユーザ名の上でダブルクリックして編集することもできます。

■ “参照”フィールドは、選択したユーザの参照番号を表わします。この番号は、ランゲージコマンドで使用されます。

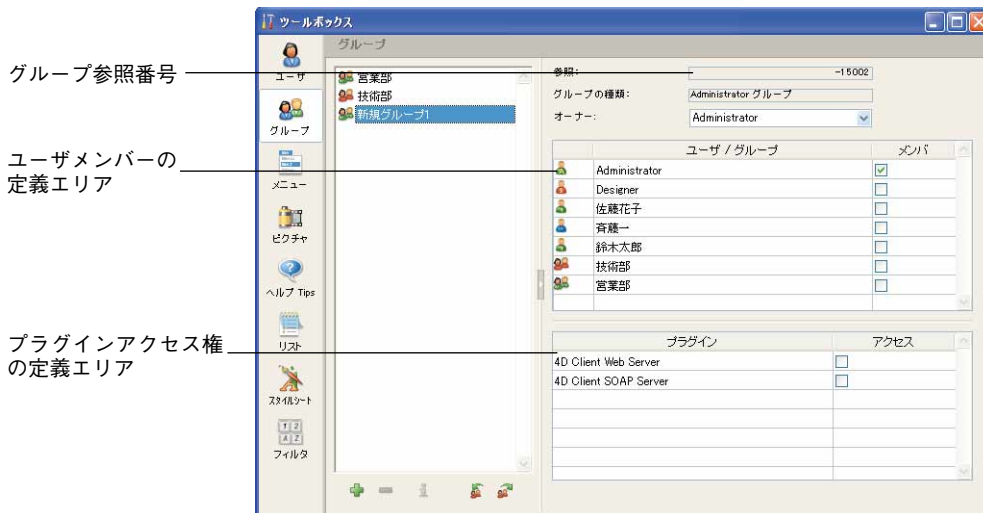
■ “ユーザの種類”フィールドは、そのユーザプロフィールの作成元を示します。次のようなユーザタイプがあります。

- Designer : Designer ユーザ
 - Administrator : Administrator ユーザ
 - Developer : Designer により作成されたユーザ
 - User : Administrator により作成されたユーザ
- “グループのメンバー” 配列には、そのデータベースで定義されたグループが表示されます。
- 「メンバー」 カラムの対応するオプションをクリックすると、選択したユーザをグループに追加することができます。また、そのオプションの選択を解除すると、ユーザをグループから削除することができます。
 - 「オーナー」 カラムは、選択したユーザがそのグループのオーナーであるかどうかを示します。このカラムは編集できません。

ユーザグループのメンバーは、「グループ」ページで定義することもできます。

グループ

このエディタを使用し、データベースのユーザグループを定義することができます。



- “参照” フィールドは、選択したグループの参照番号を表わします。この番号は、ランゲージコマンドで使用されます。
- “グループの種類” フィールドは、そのグループの作成元を示します。次のようなグループタイプがあります。
 - Designer Group : Designer により作成されたグループ

- Administrator Group：Administratorにより作成されたグループ
- Developer Group：Developerタイプのユーザにより作成されたグループ
- User Group：Userタイプのユーザにより作成されたグループ

- メンバー配列には、選択したグループを除き、そのデータベースで定義されたすべてのユーザとグループが表示されます。「メンバー」カラムのオプションは、選択したグループに属するユーザやグループを示します。

このオプションをクリックすると、選択グループからユーザやグループを追加、または削除することができます。

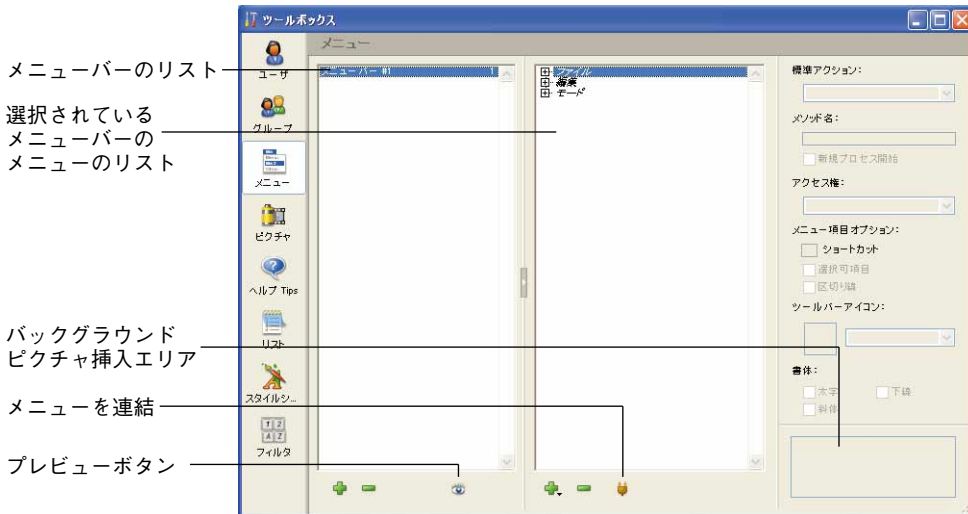
また、ユーザグループのメンバーは「ユーザ」ページでも定義することができます。


- “プラグイン”配列により、選択グループのメンバーに対して各プラグインへのアクセスを提供することができます。
- 2つの特別なボタンを使用すると、ユーザの読み込みや書き出しをディスクファイルの形式で行うことができます。データベースのAdministratorだけが、この機能を使用できます。



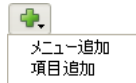
メニュー


このエディタを使用し、データベースのメニューを定義することができます。



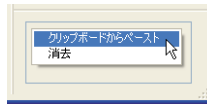
- 「プレビュー」ボタン  を使用するとプレビューモードへ切り替わり、メニューバーやメニュー、ならびにスプラッシュスクリーンが表示されます。

- メニューリストの下側にある「追加」ボタンにはポップアップメニューが関連付けられており、このメニューを使用してメニューやメニューコマンドを追加することができます。



- 選択したメニューに任意のメニューを連結するには、メニューリストの下側にある特別なアイコンをクリックした後、表示されるリストからメニューを選択します。
- 「バックグラウンドピクチャ」挿入エリア：このエリアを使用し、選択したメニューに独自のバックグラウンドピクチャを関連付けることができます。このピクチャは、「カスタムメニュー」のスプラッシュスクリーンにおいて、デフォルトで表示される青い“4D”のピクチャの代わりに使用されます。

独自のバックグラウンドピクチャをメニューに関連付けるには、ピクチャをクリップボード上に配置した後、このエリアをクリックします。すると、ポップアップメニューが表示され、クリップボードからピクチャをペーストすることができます。



「クリア」ボタンを使用すると、カスタムピクチャを削除してデフォルトのピクチャへ戻すことができます。

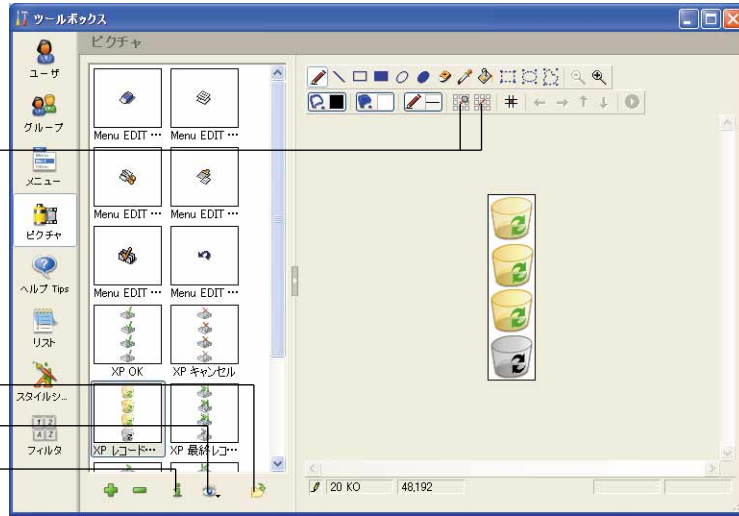
注：プレビューモードでは、ピクチャを貼り付けることはできません。

ピクチャ

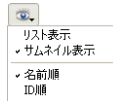
ピクチャライブラリを使用し、データベースで使用する各種ピクチャを生成することができます。

サムネール管理ウィザード
にアクセス

ピクチャプロパティの表示
ピクチャリスト管理
ピクチャの読み込み



- リストの下側にあるポップアップメニューを使用して、ピクチャリスト管理機能にアクセスすることができます。

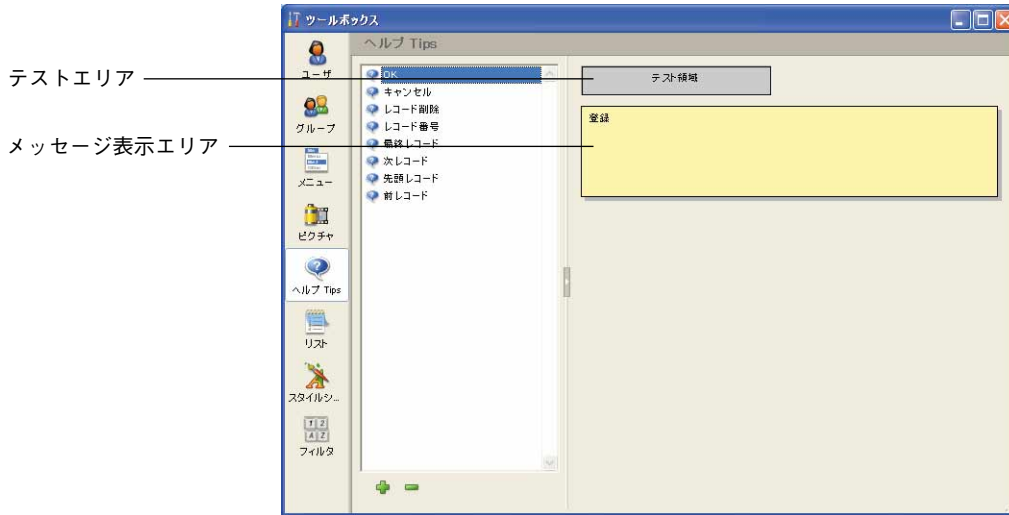


名前や番号に基づいてピクチャの並び替えを行えるようになった点に注目してください。

- ピクチャを読み込むには、📁 ボタンをクリックします。
- ウィンドウのツールバーにある 🗑️ ボタンを使い、サムネール管理ウィザード（挿入または削除）を利用することができます。

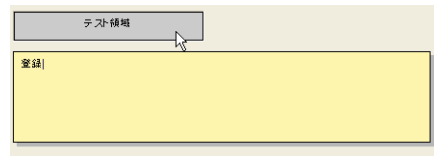
ヘルプヒント

この新しいエディタを使用し、アプリケーションのフォームで使用するヒント（Tip）の作成や管理を行うことができます。



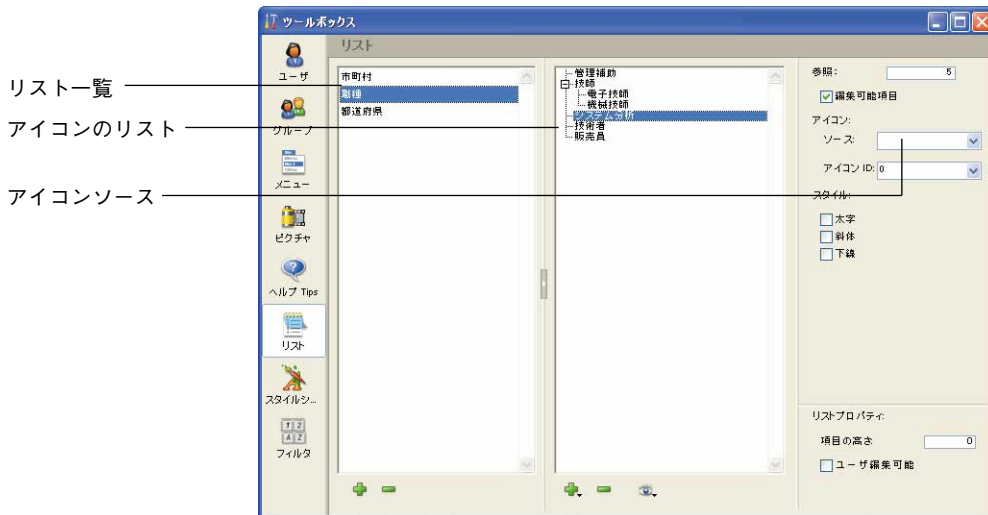
注：以前のバージョンの4th Dimension機能では、このエディタは「オブジェクトプロパティ」パレットに置かれていました。

- ヒントリストのコンテキストメニューを使用して、ヒントのリネーム、追加、削除、コピーを行うことができます。
- テストエリアを使用すると、ヘルプヒントがデータベースで表示された時の様子を見ることができます。テストを行うには、テストするヘルプヒントを選択し、クリックは行わずにマウスカーソルをテストエリアの上に置きます。



リスト

このエディタを使用すると、データベースで利用するリストを定義することができます。



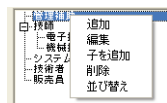
リスト一覧

アイコンのリスト

アイコンソース

このエディタで利用できる機能は、以前のバージョンの4th Dimensionのリストエディタの機能とほとんど同じです。

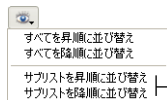
- リスト一覧のコンテキストメニューを使用して、リストのリネーム、追加、削除、コピーを行うことができます。
- さらに項目リストのコンテキストメニューには、「スレッド追加」コマンドが含まれています。



また、項目を追加するコマンドは、追加アイコンに関連付けられたメニューにも含まれています。



- 階層リストの枠内で、サブリストだけを並び替えるようになりました。

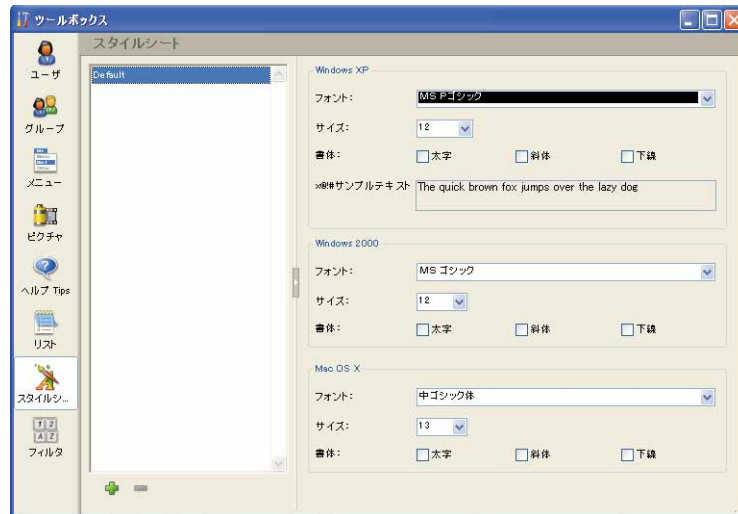


選択したサブリストを並び替えるコマンド

- 注目すべきなのは、ソースエリアの「'Cicn' resource」、 「'PICT' resource」、またはピクチャライブラリのいずれかを使用して、各項目の関連アイコンの参照先を定義できる点です。入力されるアイコン番号は、この情報を考慮します。

スタイルシート

このエディタを使用し、アプリケーションで利用するスタイルシートの作成や管理を行うことができます。



スタイルシート一覧のコンテキストメニューを使用して、スタイルのリネーム、追加、削除、コピーを行うことができます。

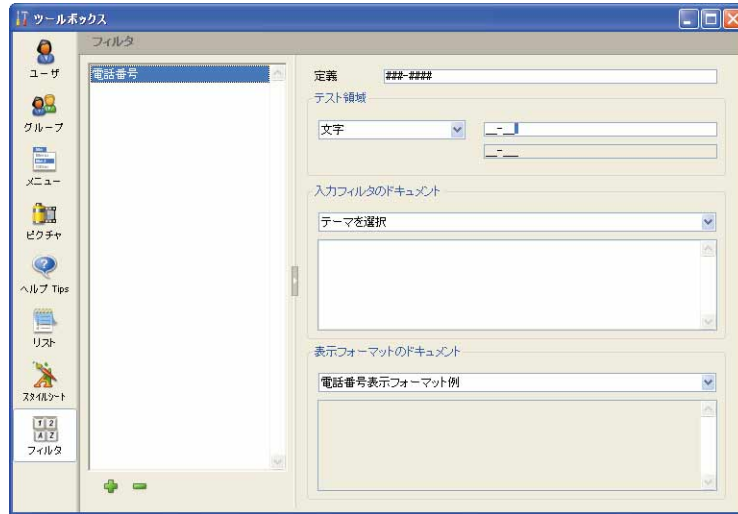
4th Dimension 2004のインタフェース管理に関する方針（後述する「プラットフォームインタフェース」の節を参照）に合わせて、各スタイルシートを3つのプラットフォーム（Windows XP、Windows 2000、Mac OS X）に対して設定できるようになります。

プラットフォームはそれぞれ、エディタの特定エリアで定義します。エリアの表示順は、現行の4th DimensionのOSに応じて変わります。例えば、4th DimensionがMac OS X上で稼働している場合、Mac OS Xエリア（ならびに関連するサンプルテキスト）がページの上部に表示されます。

このエディタの他の機能は、以前のバージョンの4th Dimensionにおけるエディタの機能と同じです。

フィルタ

このエディタを使用し、入力フィルタやカスタム表示フォーマットの作成や管理を行うことができます。



このエディタには新しいツールが用意されており、独自の表示フィルタや入力フィルタを容易に定義できるように設計されています。

- フィルタおよびフォーマット一覧のコンテキストメニューを使用して、フィルタやフォーマットのリネーム、追加、削除、コピーを行うことができます。

- 定義：フィルタやフォーマットを定義するための入力エリア。

ダブルクリックを使用して、ウィンドウの下側にあるサンプルエリアからこのエリアに文字を配置することができます。

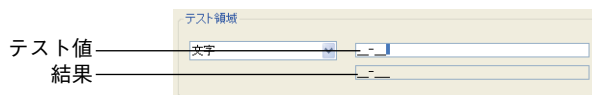
- テストエリア：このエリアを使い、定義しようとするフィルタやフォーマットの動作を制御することができます。

まず初めに、関連するメニューでフィルタやフォーマットを適用するデータタイプ（文字列、数値、日付、時間）を設定しなければなりません。

このパラメータを指定したら、関連するエリアにテスト値を入力します。

- 入力フィルタの場合、フィルタ動作の制御は入力中に行われます。

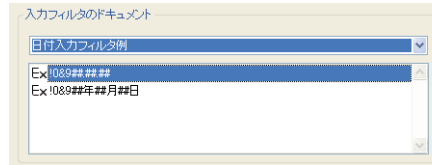
- 表示フォーマットの場合、入力が終了したところでEnterキー、またはReturnキーを押します。フォーマットの適用結果が入力エリアの下に表示されます。



- 入力フィルタのドキュメンテーション／表示フォーマットのドキュメンテーション：これらのエリアにはメニューと表示エリアがあります。

メニューを使用すると、情報やサンプルを取得したいテーマを設定することができます。

テーマを選択すると、対応する文字や情報が表示エリアに表示されます。



サンプルをダブルクリックすると、それを直接テキストエリアへ挿入することができます。

入力フィルタ

- 表示文字イニシエータ：プレースホルダの変更
- イニシエータ：フィルタ文字のイニシエータ
- 開始コード：フィルタの開始コード
- ショートハンド：フィルタのショートカット
- プレースホルダ：予定位置の設定に使用される文字
- メインデッドキャラクタ：フィルタ内で無視される文字
- 日付フォーマットのサンプル
- 時間フォーマットのサンプル
- 電話番号フォーマットのサンプル
- 保険証番号フォーマットのサンプル
- 他の入力フィルタのサンプル

表示フォーマット

- プレースホルダ：予定位置の設定に使用される文字
- メインデッドキャラクタ：フィルタ内で無視される文字
- 日付フォーマットのサンプル
- 時間フォーマットのサンプル
- 電話番号フォーマットのサンプル
- 保険証番号フォーマットのサンプル

コンパイラ

コンパイラのインターフェースが変更されましたが、ウインドウ上では今までの機能がすべて提供されます。さらに、新しいボタンを使用して「環境設定」に直接アクセスできるようになります。

「環境設定」の「コンパイラ」ページを表示する



コンパイル中の警告を無効にする

コンパイル中に、特定の警告を選択的に無効にできるようになりました。これを行うには、4Dメソッドのコードに次の命令を挿入するだけです。

```
`%W-< 警告番号>
```

警告番号は、コンパイルエラー一覧の各メッセージの最後に示されるようになりました。

例えば、次の警告を無効にする場合、

```
1: Pointer in an array declaration(518.5)
```

次のコメントを4Dメソッド、できればCOMPILER_xxxメソッド（最初にコンパイルされるメソッド）に記述する必要があります。

```
`%W-518.5
```

エクスプローラ

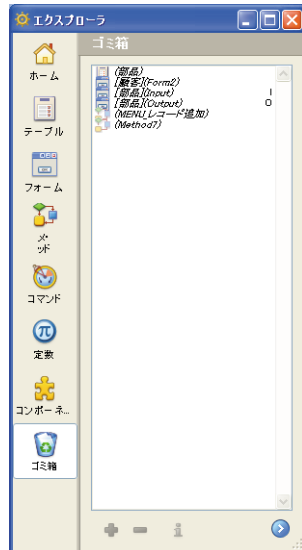
変更されたインターフェース以外にも、4th Dimension 2004のエクスプローラには新しい機能がいくつか追加されています。

- 新しい「ホーム」ページから、データベースで設定されたオブジェクトフォルダへアクセスすることができます。この新しい機能については、後述の「エクスプローラでフォルダを使用する」の節で説明しています。

- 新しい「ゴミ箱」ページでは、削除されたオブジェクトの管理を行うことができます。
- 「コンポーネント」ページにおけるプラグインやコンポーネントコマンドの詳細表示
- 新しいメソッドプレビュー機能とそのプロパティの表示

ゴミ箱ページ

「エクスプローラ」の新しい「ゴミ箱」ページには、データベースから削除されたフォルダ、テーブル、フォーム、メソッドプロジェクトオブジェクトがすべて集められます。



WindowsやMac OSのゴミ箱と同様に、4th Dimension 2004のゴミ箱はインターフェースセキュリティエレメントとして、項目を誤って削除する危険性を抑えます。

ゴミ箱内に配置されたオブジェクトは、4th Dimensionのエディタやメニュー上に表示されなくなり、修正や利用、移動などは行えません。同じような別のオブジェクトを同じ名前で作成することができます。

しかし、ゴミ箱内に配置されたオブジェクトは、ゴミ箱を空にするまでは回復可能です(後述)。

4D Server：ゴミ箱の中身はすべてのクライアントマシンで同じです。

オブジェクトの削除

「ホーム」ページ（フォルダ）または「フォーム」ページや「メソッド」ページからオブジェクトを削除することができます。

オブジェクトを削除するには、オブジェクトを選択した後、「エクスプローラ」のコンテキストメニュー（Windowsでは右クリック、Mac OSではControl+クリック）から「削除」コマンドを選ぶか、オブジェクトをゴミ箱へドラッグ、あるいはエリアの下側にある削除アイコンをクリックします。



「ホーム」ページでは、ゴミ箱もオブジェクトの移動場所として選択することができます。

また、テーブルもゴミ箱へ配置することができますが（「ホーム」ページから）、データベースにテーブルが1つしか存在しない場合を除きます。テーブルをゴミ箱へ移動すると、そのテーブルはストラクチャエディタを含めデータベースのエディタ上に表示されなくなります。しかし、データベースからテーブルを物理的に削除することはできません。テーブルを削除すると、そのテーブルのフォームは自動的に削除されます。

そのフォームの属するテーブルが削除された場合を除き、次の属性を持つフォームは削除できません：I（カレント入力フォーム）、O（カレント出力フォーム）、B（両方の属性を持つフォーム）。

最後に、これら削除不可オブジェクトのいずれかを含むフォルダも削除することはできません。

オブジェクトの回復

ゴミ箱へ移動したオブジェクトはいつでも回復することができます（ゴミ箱を空にしないかぎり）。回復を行うには、そのオブジェクトを選択してウインドウの左側の適切なページアイコンへドラッグするか、またはリストのコンテキストメニューから「移動」コマンドを選びます。



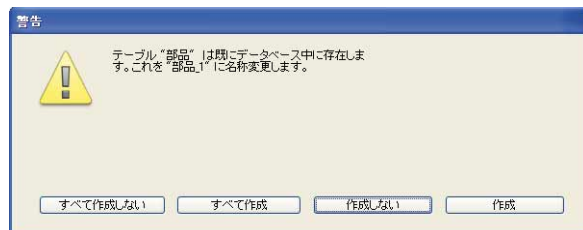
「移動」サブメニューには、データベースに存在するフォルダのリストが一覧表示されません（詳細については、後述する「エクスプローラでフォルダを使用する」の節を参照）。データベースにフォルダが存在しない場合は、「トップレベル」ラベルだけを使用できます。

このサブメニューからコマンドを選ぶと、選択オブジェクトは回復され、選択したフォルダ内へ戻されます。「トップレベル」を選択した場合、選択オブジェクトは任意のフォルダではなく第一レベルに配置されます。

1つ以上のデータベースオブジェクトの名前が、回復しようとする他のオブジェクトと同じ名前である場合、4th Dimensionより警告ダイアログボックスが表示され、次のいずれかの処理を選択することができます。

■ データベースのオブジェクトを置き換える（フォーム、メソッドなど）

■ 回復するオブジェクト名を名前_1に変更する（テーブル）



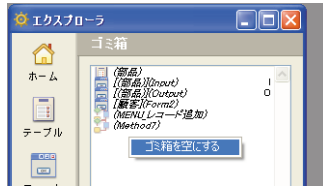
「はい」ボタンをクリックすると、回復するオブジェクトの名前を変更するか、または既存のオブジェクトを置き換えます。「すべてを変更」ボタンは、選択されたすべてのオブジェクトの名前を変更、または置き換えます（適用可能な場合）。「いいえ」または「すべて変更しない」をクリックすると、オブジェクトは回復されません。

ゴミ箱を空にする

デフォルトとして、削除されたオブジェクトは、ゴミ箱を空にするまでゴミ箱の中に残ります。例えば、圧縮の前にストラクチャファイルのサイズを縮小するために、いつでもゴミ箱を空にすることができます。ゴミ箱の中のオブジェクトはすべて、データベースから完全に削除されます。

テーブルとカレントフォーム（I、O、B属性をもつフォーム）は削除できない点に注意してください。

ゴミ箱を空にするには、エクスプローラのコンテキストメニュー（Windowsでは右クリック、Mac OSではControl+クリック）から「ゴミ箱を空にする」コマンドを選択しませんが。



注：必ずオブジェクトの外側をクリックしてください。

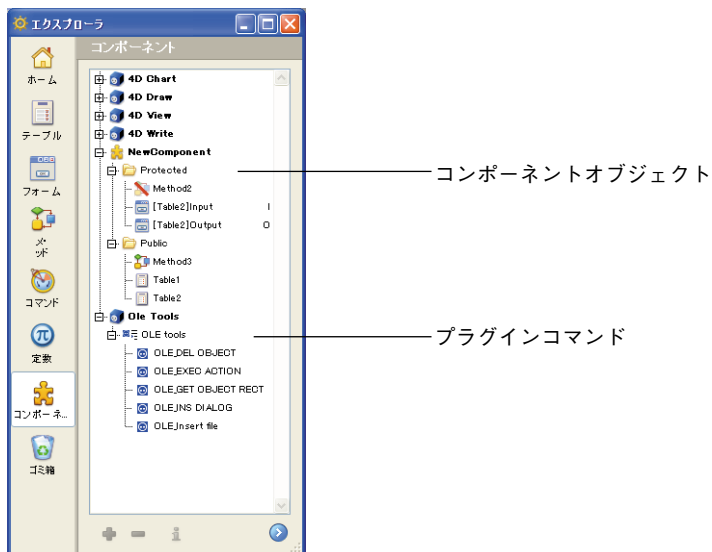
ゴミ箱内にある“削除可”のオブジェクトの数を知らせるダイアログボックスが表示され、この処理の確定またはキャンセルを実行できます。



ゴミ箱内のオブジェクトをすべて削除するには、「OK」をクリックします。

コンポーネントページの詳細表示

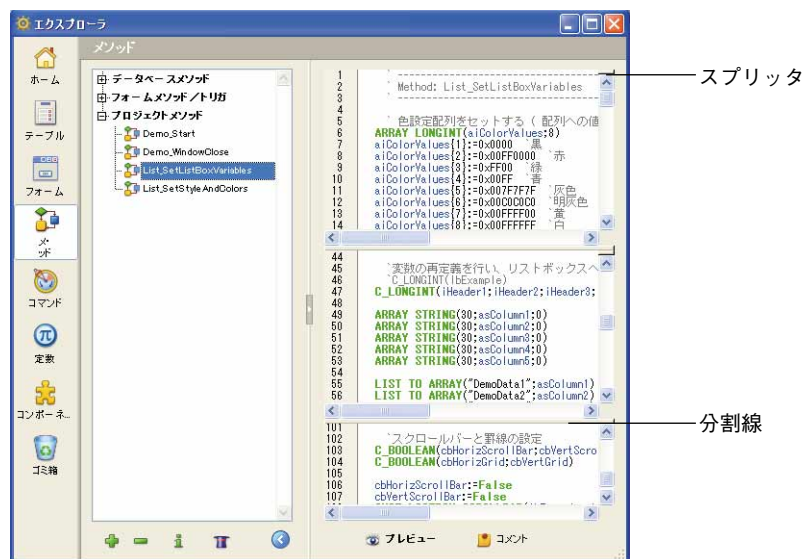
エクスプローラの「コンポーネント」ページには、データベースに含まれるプラグインとコンポーネントが詳細に表示されるようになりました。



- プラグインコマンドは、テーマ順にソートされて表示されるようになりました。これらのコマンドをドラッグ&ドロップして、メソッドに挿入することができます。
- コンポーネントオブジェクト（テーブル、フォーム、プロジェクトメソッド）は、コンポーネント順およびタイプ（パブリック、プロテクト）順にソートされ、階層リスト形式で表示されます。パブリックタイプのフォームやテーブル、メソッドは、プレビュー表示することができます。


メソッドのプレビューとプロパティの表示

メソッドのプレビューエリアには、オブジェクトに関連付けられたカラーが表示され、スクロールバー付きの独立した複数のエリアに分割できるようになり、さらにメソッドエディタのウインドウに近づきました。



エリアを追加するにはスプリッタ上でクリックします。また、エリアを削除するには、分割線をダブルクリックします。

エクスペローラに新しいアイコンが追加され、プロジェクトメソッドを選択した場合にのみ、このアイコンを使用することができます。このアイコンは、選択したメソッドの「プロパティ」ウインドウを表示します。

 —— メソッドプロパティを表示する

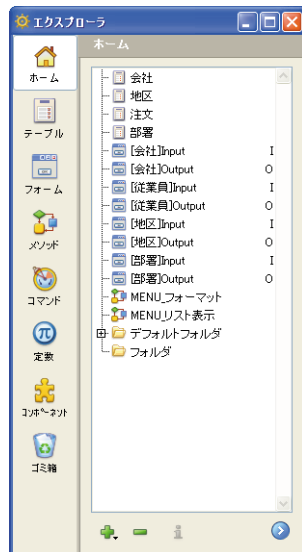
エクスプローラでフォルダを使用する

4th Dimension 2004のエクスプローラにおいて、オブジェクトフォルダを定義し、それを使用することができます。WindowsやMac OS上の便利なフォルダの実例に従って、4th Dimension 2004のフォルダを使用して、タイプ別だけではなく独自の条件に基づいて各種オブジェクトをグループ化することができます。例えば、“メッセージ”という名称のフォルダには、データベースのメッセージ機能の処理に関するオブジェクトをすべて集めることができます。

4th Dimensionでは、テーブル、フォーム、プロジェクトメソッドだけをフォルダ内に配置可能です。また、複数レベルに渡るサブフォルダを作成することもできます。

はじめに

フォルダの管理は、4th Dimensionエクスプローラの「ホーム」ページにある階層リストを用いて行います。



カスタマイズしたデータベースオブジェクト（テーブル、フォーム、プロジェクトメソッド）はすべて、常にこのページ上に表示されます。これらのオブジェクトは、エクスプローラの他のページにも表示される点に注意してください。

フォルダは階層リストの項目として表示されます。各フォルダには、あらゆるタイプのカスタマイズしたオブジェクトとその他フォルダ（サブフォルダと呼ぶ）が含まれます。

このページでは、フォルダやサブフォルダの追加、フォルダやオブジェクトのリネーム、移動、再編成を行うことができます。また、テーブルやフォーム、プロジェクトメソッドを直接作成することも可能です。エクスプローラの他のページと同様に、テーブル、フォーム、プロジェクトメソッドの名前をダブルクリックして、対応するエディタのウインドウ上で開いたり、ドラッグ&ドロップして挿入することができます。最後に、フォルダとオブジェクトは削除可能です。

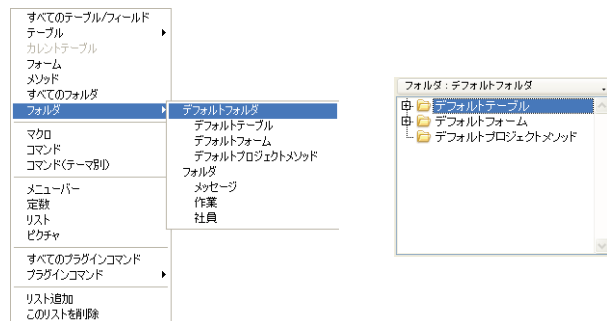
また、作成時にフォルダを各オブジェクトに割り当てることもできます。

4D Server：同じデータベースに接続したクライアントマシンはすべて、同じフォルダ構成を共有します。あるクライアントマシンが「ホーム」ページで変更を行うと（移動、追加など）、接続中のクライアントマシンすべてに対して即座に変更が自動的に反映されます。

フォルダを使用する理由

フォルダは、機能別や独自の条件別（日付、言語など）にデータベースを整理する場合に役立ちます。これにより、エクスプローラから各種オブジェクトへアクセスしやすくなります。

また、4th Dimension のメソッドエディタでは、階層リスト形式でフォルダを利用することができます。これにより、メソッドを作成する際に必要なオブジェクトを呼び出しやすくなります。



最後に、4th Dimension のフォルダは 4D Insider のグループと互換性があります（逆の場合も同様）。したがって、2つのデータベース間ですべての機能を簡単に移動することができます。この件に関する詳細は、後述の「4D Insider のグループを使って作業する」の節を参照してください。

従来の動作との互換性

フォルダは、データベースオブジェクトの新しい分類方法を提供するものであり、従来のオブジェクト階層構造に代わるものではありません。また、使用しなくても構いません。フォルダを活用するには、エクスプローラの「ホーム」ページを使用します。ただし、テーブル、フォーム、メソッドなどのページは従来のように使用可能であり、以前のバージョンの4th Dimensionと同様に動作します。

デフォルトフォルダ

フォルダが存在しないデータベースの場合（以前のバージョンの4th Dimensionから変換されたデータベース、または新しいデータベースでユーザがフォルダを作成しない場合）、4th Dimensionは“デフォルトフォルダ”を作成し、この中に「デフォルトフォーム」、「デフォルトプロジェクトメソッド」、「デフォルトテーブル」という名称の3つのサブフォルダを作成します。

■ 新規データベースの作成時に、デフォルトフォルダは作成されません。ユーザがフォルダを作成しない場合、4th Dimensionはデータベースの次回起動時にデフォルトフォルダと各サブフォルダを作成し、そこにユーザが作成したオブジェクトを配置します。

ユーザが少なくとも1つのフォルダを作成した場合、プログラムはデフォルトフォルダを作成しません。

■ 以前のバージョンの4th Dimensionで作成され、バージョン2004に変換されたデータベースを開くと、4th Dimensionはデフォルトフォルダを作成し、そこにすべてのオブジェクトを配置します。ただし、4D Insiderのグループが定義されている場合、4th Dimensionは何も行いません。

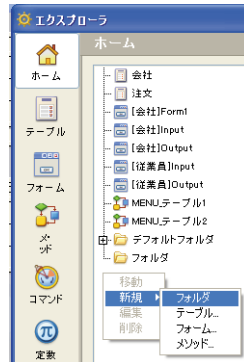
このデフォルトフォルダの目的は、既存のオブジェクトを格納することであり、そのサブフォルダは特別なプロパティを一切もちません。これらのフォルダの変更、削除、リネームは任意に行うことができます。

次に、データベースで新たに作成したオブジェクトは、デフォルトではフォルダリストの最上位レベル、つまりフォルダの外側（“トップレベル”）に配置されます。ユーザは自由に必要なフォルダを作成したり、その内容を管理することができます。

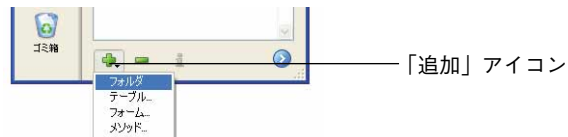
フォルダやサブフォルダを作成する

エクスプローラの「ホーム」ページ上で、フォルダやサブフォルダをいつでも追加することができます。

これを行うには、このエリアのコンテキストメニュー（Windowsでは右クリック、Mac OSではControl+クリック）から「新規>フォルダ」コマンドを使用します。



また、このエリアの下側にある「追加」アイコンをクリックすると表示されるメニューを使用することもできます。



フォルダまたはサブフォルダは即座に作成され、“フォルダ+フォルダ番号”という名前がデフォルトで付けられます。この名前は自由に変更することができますが、フォルダやサブフォルダの名前は重複してはいけません。

- フォルダを作成するには、エリア内に既存するすべてのフォルダの外側をクリックするか、項目が何も選択されていないことを確認してから「追加」アイコンをクリックします。
- サブフォルダを作成するには、サブフォルダを追加しようとするフォルダをクリックするか、フォルダを選択した後に「追加」アイコンをクリックします。

次に、ドラッグ&ドロップや移動コマンドを使用して、フォルダをサブフォルダに変更したり、またはその逆の作業を簡単に行うことができます（後述）。

フォルダやオブジェクトの名前を変更する

フォルダ（または「ホーム」ページ上の任意のオブジェクト）の名前は、いつでも変更することができます。これを行うには、その名前の上でAlt+クリック（Windows）またはOption+クリック（Mac OS）します。また、項目上でクリックを二度行う方法もあります（実質上のダブルクリックとならないように、2回のクリックの間は少し時間を置く）。

すると項目が編集可能になります。



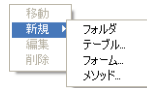
変更後、エリアの外側をクリックするか、Tab キーを押して入力を確定します。

注

- ・フォルダ名の長さは31桁までです。
- ・同じ名前を2つのフォルダに付けることはできません（階層レベルに関係なく）。
- ・フォーム名の先頭に置かれたテーブル名を変更しても、テーブルの名前を変えることはできません（例：[MyTable]Form）。

オブジェクトの作成と変更

コンテキストメニューの「新規」コマンドを使用すると、フォルダだけではなく（前述した「フォルダやサブフォルダを作成する」の節を参照）、テーブルやフォーム、プロジェクトメソッドも作成することができます。



このタイプのサブコマンドを選択すると、オブジェクト作成用のダイアログボックスへ直接アクセスします（後述の「作成中にオブジェクトをフォルダへ追加する」を参照）。

コンテキストメニューの「編集」コマンドを使用すると、選択したオブジェクトを適切なエディタウインドウ（テーブル、フォーム、プロジェクトメソッド）に表示することができます。



選択した項目がフォルダやサブフォルダである場合、または複数の項目を選択した場合には、このコマンドがグレー表示されます。

オブジェクトの削除

「削除」コマンドを使用すると、選択項目が削除されます。選択項目内にフォルダやサブフォルダが含まれる場合、その中に納められた項目はすべて削除されます。



このコマンドは、あらゆるタイプの項目（フォルダ、サブフォルダ、テーブル、フォーム、プロジェクトメソッド）に対して使用可能であり、また複数項目を選択することもできます。削除されたオブジェクトは、リストおよびアプリケーションのエディタ上には存在しなくなります。

削除された項目は、エクスプローラの「ごみ箱」内に置かれます。エクスプローラの「ごみ箱」の機能に関する詳細は、前述の「ごみ箱ページ」を参照してください。

また、エクスプローラの「ホーム」ページからオブジェクトを1つ以上削除するには、オブジェクトを選んだ後に、このエリアのコンテキストメニュー（Windowsでは右クリック、Mac OSではControl+クリック）の「移動>ごみ箱」コマンドを選択することもできます。

フォルダとサブフォルダを整理する

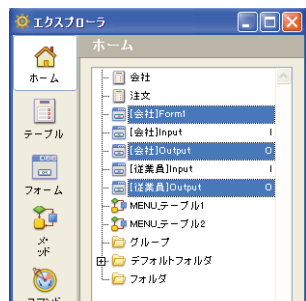
フォルダやサブフォルダの内容は、いつでも再編成することができます。オブジェクトをフォルダ内に配置したり、フォルダを別のフォルダに納めることができます。フォルダを別のフォルダに配置すると、それはサブフォルダになります。フォルダとサブフォルダの特性は同じです。

フォルダの移動は、その中の項目とともに行われます。移動後のオブジェクトの階層構造は、新しい場所でも以前の状態が維持されます。

単一オブジェクト、または複数のオブジェクトを移動することができます。

連続するオブジェクトを選択したい場合、Shiftキーを押しながら最初と最後のオブジェクトをクリックします。

連続していないオブジェクトを選択したい場合、Ctrlキー（Windows）またはCommandキー（Mac OS）を押しながら、各オブジェクトをクリックします。



フォルダ内容を整理するには、次の3通りの方法があります。

■ ドラッグ&ドロップを使用する。

- コンテキストメニューの移動コマンドを使用する。
- 各オブジェクトの作成時に「フォルダ」メニューを使用する。

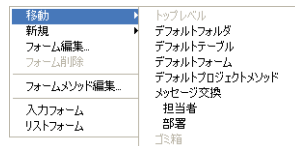
ドラッグ&ドロップ

エクスプローラの「ホーム」ページでは、ドラッグ&ドロップを使用してフォルダ内容を整理することができます。フォルダ、オブジェクト、選択項目を移動するには、クリックしてマウスボタンを押したまま項目をドラッグし、それを別の場所にドロップします。

移動

「ホーム」ページのコンテキストメニューの「移動」コマンドを使用すると、最初に目的のフォルダを展開しなくても、フォルダ内容を変更することができます。

これを行うには、オブジェクトやフォルダ、選択項目上でマウスボタンを右クリック (Windows) または Control+クリック (Mac OS) して、コンテキストメニューを表示します。階層形式の「移動」コマンドは、移動先となりうる既存のフォルダやサブフォルダをすべて提示します。



選択項目を配置しようとするフォルダを選択します。「トップレベル」を選択すると、選択項目はリストの第一レベルに置かれ、あらゆるフォルダの外側に配置されます (既に第一レベルに位置するフォルダを移動しようとする、この行はグレー表示されます)。

「ごみ箱」を選択すると、選択項目が削除されます (詳細については、前述の「オブジェクトの削除」の節を参照)。

オブジェクトの作成時にフォルダへ追加する

新規オブジェクトの作成時に、オブジェクトをフォルダへ直接配置することができます。テーブル、フォーム、プロジェクトメソッドを作成するダイアログボックスの新しいメニューを使用すると、オブジェクトを作成するフォルダを設定することができます。デフォルトでは、第一レベル (トップレベル)、あるいは前回エクスプローラで選択したフォルダ内 (適用可能な場合) にオブジェクトが作成されます。

- 新規テーブル作成ダイアログボックス



テーブルが追加されるフォルダの選択メニュー

■ 新規フォームウィザード



フォームが追加されるフォルダの
選択メニュー

■ 新規プロジェクトメソッド作成ダイアログボックス



メソッドが追加されるフォルダの
選択メニュー

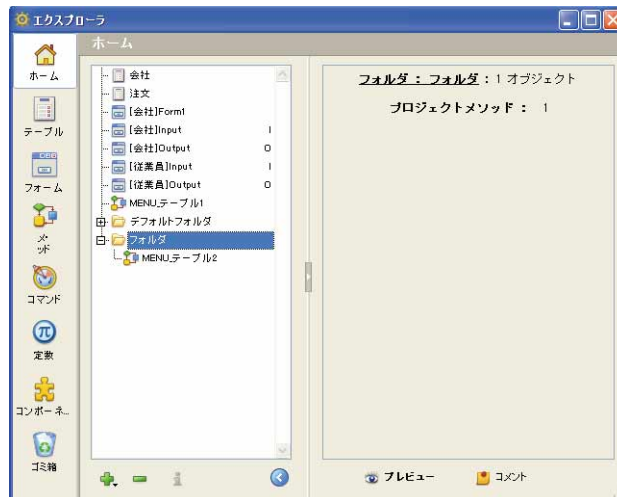
もちろん、作成した後はエクスプローラの「ホーム」ページにおいて、これらのフォルダオブジェクトをいつでも変更することができます。

また、エクスプローラの「ホーム」ページからコンテキストメニューや「追加」アイコンを使用し、新規オブジェクト（テーブル、フォーム、プロジェクトメソッド）の作成を直接要求できる点に注目してください。



フォルダ情報

エクスプローラでフォルダを選択すると、ウインドウのプレビューエリアには複数の情報が表示されます。



- フォルダ: フォルダ名: フォルダ内に存在するオブジェクトの数
- テーブル: フォルダ内に含まれるテーブルの数
- フォーム: フォルダ内に含まれるフォームの数
- プロジェクトメソッド: フォルダ内に含まれるプロジェクトメソッドの数
- フォルダ: フォルダ内に含まれるフォルダ (サブフォルダ) の数

あるオブジェクトタイプがフォルダ内に存在しない場合、そのタイプはプレビューエリアに表示されません。フォルダが空の場合、このエリアには「フォルダ:フォルダ名」とだけ表示されます。

複数のフォルダやオブジェクトを選択すると、カレント項目、つまり前回クリックした項目に関する情報が表示されます。

4D Insider のグループを使って作業する

4th Dimension で作成したすべてのフォルダは、4D Insider においてグループとして表示されます（逆の場合も同様）。また、同じ操作を適用することができます。実際、これら2つの構造は類似しています。

しかし、4D Insider で作成したグループには、他のタイプのオブジェクトを格納することができます。つまり、メニューバー、メニュー、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、（フィールドにリンクしていない）ヒント、データベースメソッド、STR#リソースなどのオブジェクトです。

これらのオブジェクトは4th Dimension のエクスプローラ上に表示されません。したがって、4th Dimension でフォルダやグループを削除しても、そこに含まれる可能性がある非表示オブジェクトは削除されません。これらのオブジェクトは、自動的にフォルダから取り出され、依然として他のエディタから利用することができます。4D Insider 側では、これらのオブジェクトはデータベースの第一階層（グループの外側）に配置されます。

一方で、特定のオブジェクトは、その“親”オブジェクトから切り離すことができず、4D Insider でしか表示されなくても、その関連オブジェクトと一緒に削除されます。

- フォームメソッドはそのフォームと一緒に削除されます。
- オブジェクトメソッドはそのフォームと一緒に削除されます。
- フィールドにリンクしたヒントはテーブルと一緒に削除されます。
- トリガはテーブルと一緒に削除されます。

メソッドエディタ

4th Dimension 2004 のメソッドエディタには、主要な3つの機能が新たに追加されました。

- メソッドを呼び出すことができる、メソッドマクロコマンド
- マクロフォルダ管理
- メソッドエディタウインドウの新しいツールバー

4D コードマクロの呼び出し

4th Dimension 2004のメソッドエディタを使用して、4Dのプロジェクトメソッドを実行するマクロコマンドの生成を行い、それを利用することができます。これにより、開発者はコンポーネントに関連付けたマクロコマンドを使用し、洗練された機能を作成して配付することができます。

<method> タグ

メソッドから4Dコードを呼び出すため、4th Dimension 2004のマクロ言語に新しいダブルタグ“<method> </method>”が追加されました。

これらのタグは、プロジェクトメソッド名の開始と終了ならびにその引数（任意）を表わします。このマクロがコールされると、メソッドが実行されます。引数は“(“param1;param2;...”)”という形式で渡すことができます。メソッドでは\$1、\$2などの変数を使用し、この引数を受け取ります。例えば、次のマクロはカレントメソッド名を引数として使用し、MyMethodメソッドの実行を開始します。

```
<method>MyMethod("<method_name/>")</method>
```

呼び出されたメソッドの実行コンテキスト

呼び出されたメソッドのコードは、新規プロセスで実行されます。このプロセスは、メソッドが実行されると終了します。

ストラクチャプロセスは、呼び出されたメソッドの実行が終了するまで中断します。したがって、メソッドは必ず短時間で終了し、アプリケーションをブロックする危険性がないよう注意してください。問題が発生した場合、*Ctrl+F8* (Windows) や *Command+F8* (Mac OS) を使用して、プロセスを終了してください。

呼び出されたメソッドにおいて、4th Dimensionが自動で管理する特種なプロセス変数を使用することができます。特定の変数は、4th Dimensionにより値が代入されますが（入力変数）、他の変数は開発者が値を設定しなければなりません（出力変数）。

入力変数	タイプ	説明
_textSel	C_TEXT	<-- 選択されたテキスト (32000を超える場合は空)
_blobSel	C_BLOB	<-- 選択されたテキスト
_selLen	C_LONGINT	<-- 選択されたテキストの長さ
_textMethod	C_TEXT	<-- メソッドテキスト (32000を超える場合は空)
_blobMethod	C_BLOB	<-- メソッドテキスト
_methodLen	C_LONGINT	<-- メソッドテキストの長さ

出力変数	タイプ	説明
<code>_textReplace</code>	<code>C_TEXT</code>	--> 代替テキスト、デフォルトは""
<code>_blobReplace</code>	<code>C_BLOB</code>	--> 代替テキスト、デフォルトは""
<code>_action</code>	<code>C_LONGINT</code>	--> メソッド呼び出しにおける置換タイプ 0= なし 1= <code>_textReplace</code> を挿入 2= <code>_blobReplace</code> を挿入 3= <code>_textReplace</code> でメソッドを置換 4= <code>_blobReplace</code> でメソッドを置換

例題

このマクロコマンドの例題は、選択したテーブルにレコードを作成し、そのタイプに関わらずレコードの各フィールドを初期化するコードを自動的に挿入します。このマクロは `Init_macro_record` メソッドを呼び出しますが、このメソッドはデータベースから使用できなくてはなりません。

■ マクロコマンドのコードは次の通りです。

```
<macro name="Init_record">
<text>
<method>Init_macro_record</method>
</text>
</macro>
```

■ `Init_macro_record` プロジェクトメソッドのコードは次の通りです。

```
C_TEXT(_textSel;_textReplace)
C_TEXT($text)
$text:=_textSel

$start:=Position("[";$text)
If($start>0)
    $text:=Substring($text;$start)
    $end:=Position("]";$text)
    If($end>0)
        $text:=Substring($text;1;$end)
        $table:=0
        For($i;1;Count tables)
            If($text=("["+Table name($i)+"])")
                $table:=$i
                $i:=Count tables+1
            End if
```

End for

\$br:=**Character**(Carriage return)

\$to_insert:=**Command name**(68)+"("\$text+")" + \$br

`68= **CREATE RECORD** コマンド

For(\$i;1;**Count fields**(\$table);1)

\$field:=\$text+**Field name**(\$table;\$i)

\$type:=**Type**(**Champ**(\$table;\$i)->)

Case of

¥ (\$type=Is Alpha Field) | (\$type=Is Text)

\$to_insert:=\$to_insert+\$field+":=¥"¥"+\$br

¥ (\$type=Is Real) | (\$type=Is Integer) | (\$type=Is Longint)

\$to_insert:=\$to_insert+\$field+":=0"+\$br

¥(\$type=Is Date)

\$to_insert:=\$to_insert+\$field+":=!00/00/00!" + \$br

¥ (\$type=Is Time)

\$to_insert:=\$to_insert+\$field+":=?00:00:00?" + \$br

¥ (\$type=Is Boolean)

\$to_insert:=\$to_insert+\$field+":="

+**Command name**(215)+\$br

`215= **False** コマンド

¥ (\$type=Is Picture)

\$to_insert:=\$to_insert+**Command name**(286)

+"(\$pict)" + \$br

`286= **C_PICTURE** コマンド

\$to_insert:=\$to_insert+\$field+":=\$pict" + \$br

¥ (\$type=Is BLOB)

\$to_insert:=\$to_insert+**Command name**(604)

+"(\$blob)" + \$br

`604= **C_BLOB** コマンド

\$to_insert:=\$to_insert+\$field+":=\$blob" + \$br

End case

End for

_textReplace:=\$to_insert+**Command name**(53)+"("\$text+")" + \$br

`53= **SAVE RECORD** コマンド

_action:=1

Else

_action:=0

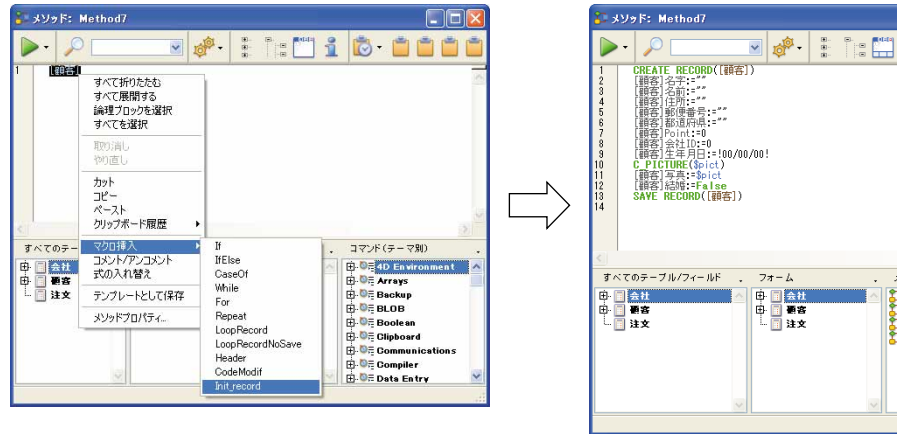
End if

Else

_action:=0

End if

■ 次は、このマクロを図で示したものです。



新しいマクロフォルダ

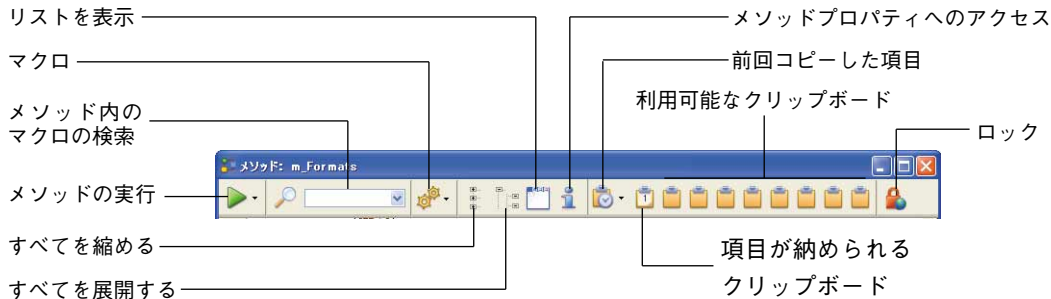
4th Dimension 2004は、複数のマクロドキュメントをXML形式で受け入れます。これらのドキュメントは、必ず「Macros」という名前のフォルダ内に配置し、「Macros.xml」ファイル（そのマシンのアクティブな4Dフォルダ内に存在）と同じ階層に置かなければなりません。

「Macros.xml」ファイルは以前のものと同一機能を持ちます。まず初めに、4th Dimensionは「Macros.xml」ファイルのマクロコマンドを処理し、次に「Macros」フォルダ内にあるマクロコマンドを処理します（利用可能な場合）。

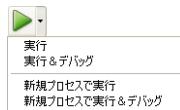
この仕組みにより、4th Dimensionアプリケーションで複数のマクロファイルを同時に使用することができます。また、カスタマイズしたマクロコマンドの配付も容易になります。

新しいツールバー

メソッドエディタの各ウインドウにはツールバーが用意され、メソッドの実行や編集に関する基本的な機能を簡単に利用できるようになります。



- **メソッドの実行**：このボタンは、「デザイン」モードでメソッドの実行を開始します。このボタンに付属するメニューを使用すると、実行タイプを選択することができます。



- **検索**：検索アイコンは、標準の検索ダイアログボックスを表示します。このダイアログボックスを使用し、カレントメソッド内で検索を実行することができます。
- **マクロ**：このボタンは、ウインドウ上で使用可能なマクロコマンドをすべて一覧するメニューを表示します。
- **前回の項目**：このボタンは、ウインドウ上でコピーされた項目のうち、直前までの20項目を一覧するメニューを表示します。項目を選択すると、カーソルが置かれている位置にその項目がコピーされます。
- **クリップボード**：これら9つのアイコンは、メソッドエディタ上で利用できる9つのクリップボードを示します。数字を含む白いアイコンはクリップボードにデータが含まれていることを表わし、茶色のアイコンはクリップボードが空であることを表わします。
- **テキストをクリップボードへコピーするには**、メソッド内でテキストを選択し、コピー先となるクリップボードのアイコン上で **Shift+クリック** するか、**Ctrl+Shift+クリップボード番号 (Windows)** または **Command+Shift+クリップボード番号 (Mac OS)** を押します。アルファベット用キーボード上の数字キーを使用しないでください。

■ クリップボードの内容をカーソル位置にペーストするには、そのアイコンをクリックするか、Ctrl+クリップボード番号 (Windows) または Command+クリップボード番号 (Mac OS) を押します。

アルファベット用キーボード上の数字キーを使用しなくてはならない点に注意してください。

■ すべて展開する／すべて折りたたむ：このボタンを使用して、メソッドの制御フロー構造をすべて展開、または縮めることができます。

■ リストの表示：このボタンを使用して、ウインドウ上のリストを表示したり、または隠すことができます。

■ ロック：このボタンは、クライアント／サーバにおいてメソッドが他のユーザによりロックされていることを示します。南京錠のアイコンをクリックすると、メソッドを再ロードすることができます (これにより、他のユーザによる変更を表示)。

■ プロパティ：このボタンは、「メソッドプロパティ」ダイアログボックスを表示します。

フォームエディタ

製品のインタフェースに加えて、4th Dimension 2004の「フォーム」エディタの機能がいくつか変更されています。

■ 「オブジェクトプロパティ」パレットの削除、ならびに「プロパティリスト」パレットの変更

■ 各ウインドウに統合された新しいオブジェクトやツールバー

■ フォームの印刷設定定義に関する変更

■ 新しいツール：ズーム、表示、バッジ、オブジェクトテンプレート、新しいマグネティック機能

パレットとプロパティリスト

4th Dimension 2004では、「オブジェクトプロパティ」パレットが削除されています。このパレットの情報と機能はすべて、「プロパティリスト」において利用することができるため、このパレットが不要になりました。

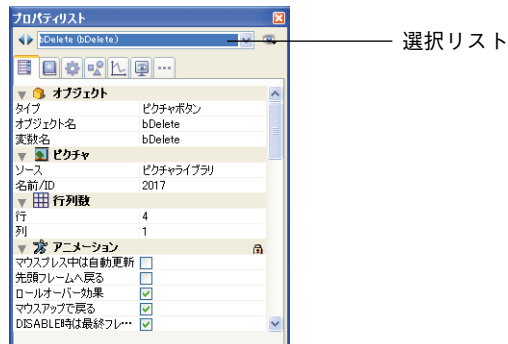
「オブジェクトプロパティ」パレット (4th Dimension 2003)



さらに、4th Dimension 2004の「プロパティリスト」には新しい機能が提供され、選択リスト内容の選択や並び替えを行えるようになります。また、コンテキスト表示とプロパティテーマのいずれかを選択することも可能です。

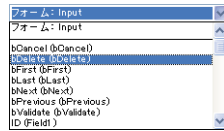
選択リストの設定

選択リストを使用し、プロパティを表示しようとするオブジェクトを選択することができます。

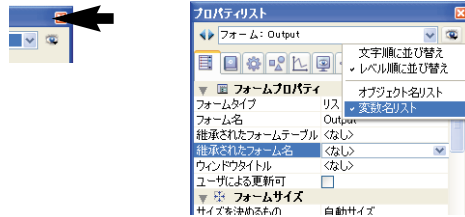


4th Dimension 2004において、このリストは3つの部分に分かれており、ラインで区切られています。

- フォーム名
- ダイナミックオブジェクト (変数にリンク)
- スタティックオブジェクト (変数へのリンクなし)



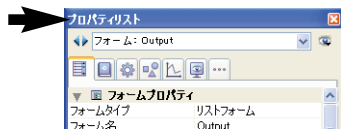
各ダイナミックオブジェクトに関し、このリストには変数名の後ろにオブジェクト名が括弧で括られて表示されます（その逆も同様）。リストに関連付けられたメニューを使用して、並び順を指定することができます。



コマンドは、最初に表示される名前のタイプを示します。

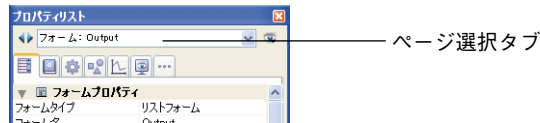
また、このリストの各パートの並び順を変更することもできます。デフォルトとして、リストはアルファベット順にソートされています。関連付けられたメニューを使用し、レベル順に従って並び替えを行うことも可能です（前面から背面へ）。


リストの左側にあるボタンを使用すると、リスト上の各オブジェクトを上から下へ順に選択することができます。



プロパティ表示ページ

ページ選択タブを使用して、「プロパティリスト」に表示される情報を選択することができます。




- 最初のページである  「メイン」は、以前のバージョンの4th Dimensionの「プロパティリスト」と同様の機能があります。つまり、各種テーマ（データソース、座標やサイズ、イベントなど）を用いて、選択されたオブジェクトに関するあらゆる情報を表示します。


その表示は前後関係に依存します。つまり、選択したオブジェクトのタイプに関するテーマやプロパティだけが表示されます。


また、特定のプロパティの表示内容は、他のプロパティの値に応じて動的に変更されます。


- 他の6つのページにはそれぞれ、選択したオブジェクトに関する特定情報が含まれています。また、オブジェクトタイプに関わらず、各テーマのプロパティがすべて表示され、関係のないプロパティはグレー表示されます。


このタブ機能により、特定オブジェクトのプロパティをより全般的に設定できるようになります。次に、6つのテーマページについて説明します。


-  **設定**：オブジェクトの特性を定義するプロパティを含みます（タイプ、名前、データソースなど）。

-  **アクション**：オブジェクトのダイナミック動作を定義するプロパティを含みます（関連メソッド、ドラッグ&ドロップ、フォームイベント）。

-  **サイズ**：座標、サイズ、オブジェクトリサイズを定義するプロパティを含みます。

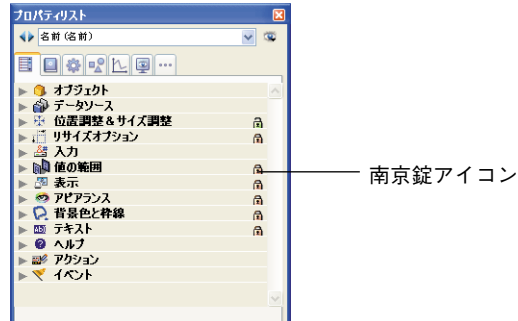
-  **値**：入力可オブジェクトの編集や表示を定義するプロパティを含みます（入力可、ショートカット、表示フォーマットなど）。

-  **アピアランス**：オブジェクトの外観を定義するプロパティを含みます。

-  **仕様**：選択したオブジェクトタイプのプロパティを含みます（フィールドのヘルプメッセージ、アニメーションボタン、フォームの印刷設定など）。他のテーマページと異なり、そのオブジェクトと関係のないプロパティは表示されません。

南京錠アイコン

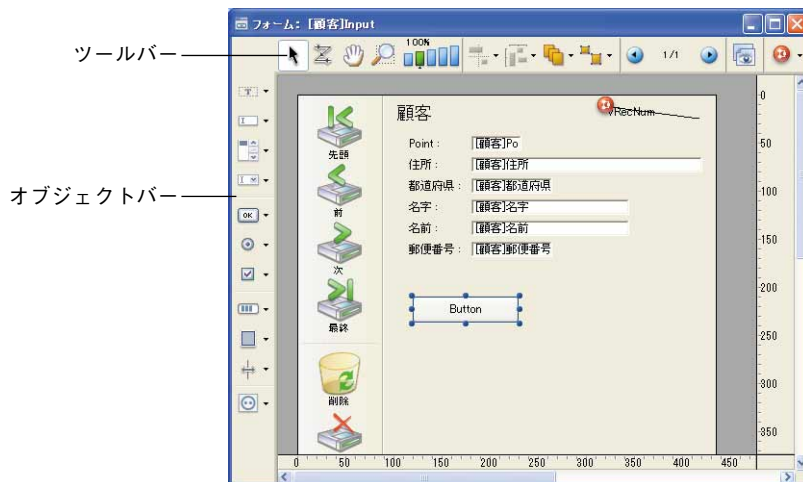
フォームに対して「ユーザ編集可」プロパティが適用されている場合、プロパティリストには南京錠アイコンが表示され、ユーザがそのプロパティを変更可能かどうかを示します。



詳細については、後述の「ユーザフォーム」の節を参照してください。

統合されたオブジェクトバーとツールバー


4th Dimension 2004では、オブジェクトパレットとツールパレットがひとつのバーとして「フォーム」エディタの各ウィンドウに組み込まれています。




より人間工学に基づいたこの新しい機能を使用し、オブジェクトや主要な機能へダイレクトにアクセスすることができます。


ツールバー

新しいツールバーには、次の項目があります。


 **ユーザロック**：ツールバーの左側にこのアイコンが表示されている場合、そのフォームが“ユーザ編集可”であることを示します。この場合、直接そのフォームを編集することはできないので、まず初めにこのアイコンをクリックしてロックを解除しなくてはなりません。詳細については、後述の「ユーザ編集可」の節を参照してください。

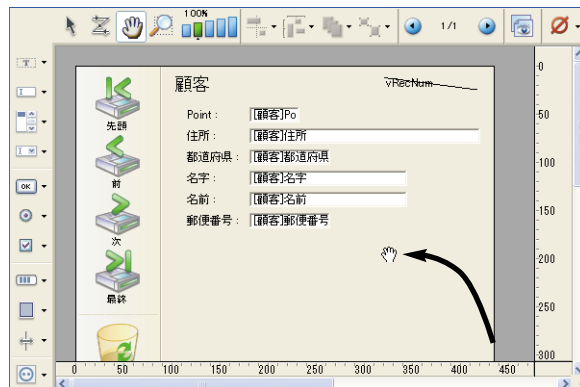
 **選択ツール（機能は変わらず）**：フォームオブジェクトの選択、移動、サイズ変更を行うことができます。

注：テキストやグループボックスタイプのオブジェクトを選択した場合、Enter キーを押すと編集モードへ移行することができます。


 **入力順序（機能は変わらず）**：“入力順序”モードへ切り替わり、フォームの現在の入力順序を表示したり、変更することができます。

注：「バッジ」により、フォームの作業を続けながら、現在の入力順序を表示することができる点に注意してください。


 **移動**：「移動」モードへ移行します。このモードでは、ウインドウ上でドラッグ&ドロップを使用し、フォームのあらゆる場所へ素早く到達することができます。カーソルは手のひらの形に変わります。




この移動モードは、特にフォーム上でズームを行う場合に役立ちます。

 **ズーム**：フォームの表示倍率を変更することができます（デフォルトは100%）。


虫めがねのボタンをクリックするか、目的のバーを直接クリックすると、「ズーム」モードに移行することができます。この新しい機能については、後述の「ズーム」の節で説明しています。

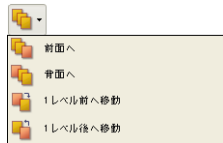
 **整列（機能は変わらず）**：このボタンは、フォーム上のオブジェクトの整列を行うメニューにリンクしています。




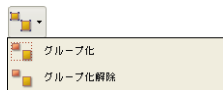
 **均等配置ボタン（機能は変わらず）**：このボタンは、フォーム上のオブジェクトの均等配置を行うメニューにリンクしています。




 **レイヤボタン（機能は変わらず）**：このボタンは、フォーム上のオブジェクトの階層を変更するメニューにリンクしています。



 **グループ化／グループ解除（機能は変わらず）**：このボタンは、フォーム上の選択オブジェクトのグループ化やグループ解除を行うメニューにリンクしています。





注：グループに属するオブジェクトは、そのセット全体のグループ化を解除しなくても選択できるようになります。これを行うには、そのグループを選択し、次にオブジェクト上でCtrlキー+クリック（Windows）またはCommandキー+クリック（Mac OS）します。

 **表示とページ管理**：このエリアを使用して、あるフォームページから別のページへ移動したり、ページを追加することができます。フォームページを移動するには、矢印ボタンをクリックするか、中央のエリアをクリックすると現われるメニューから、表示するページを選択します。



最終フォームが表示されている状態で、右矢印ボタンをクリックすると、4th Dimension はページの追加を行います。

 **ビュー管理**：このボタンは、ビューパレットの表示や非表示を管理します。この新しい機能については、後述の「ビューの管理」の節で説明しています。

 **バッジ表示**：このボタンをクリックするたびに、すべてのタイプのフォームバッジが連続して表示されます。また、このボタンは表示するバッジのタイプを直接選択できるメニューにリンクしています。



この新しい機能については、後述の「バッジの使用」の節で説明しています。

オブジェクトバー

新しいオブジェクトバーには、4th Dimension のフォーム上で使用できるオブジェクトがすべて含まれています。

あるオブジェクトタイプを描画するには、対応するボタンを選択した後、フォーム上でそのオブジェクトを描きます。

また、ボタンを2回クリックすると、フォーム上でオブジェクトを描画した後も、そのボタンは選択されたままになります。この機能により、同じタイプのオブジェクトをいくつも連続して作成しやすくなります。



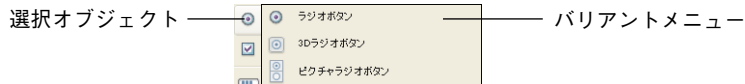
選択オブジェクト



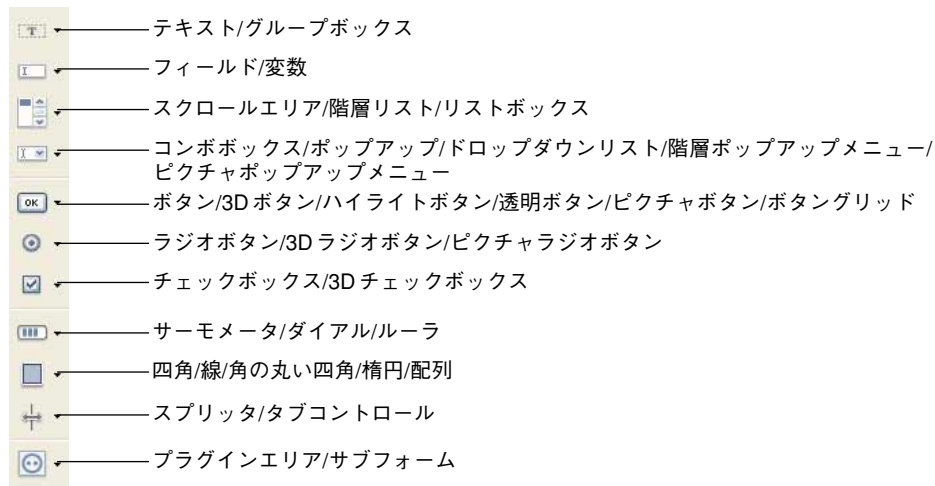
選択継続

注：選択継続をキャンセルするには、別のツールをクリックします。

最後に、ボタンの右側をクリックすると、バリエーションメニューにアクセスします。



4th Dimension の前リリース以降、オブジェクトのテーマ分けが変更され、新しい機能が追加されています。次に、4th Dimension 2004 で利用可能なオブジェクトファミリーを示します。



注：新しいフォームオブジェクトの機能については、後述する「フォームオブジェクト」の節で説明しています。

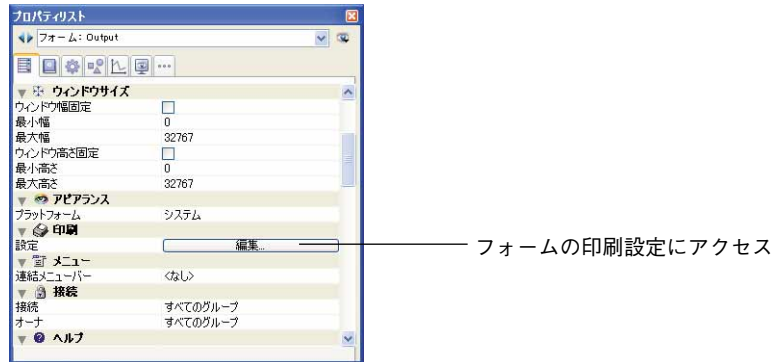
フォームの印刷設定を編集する

4th Dimension 2004 では、フォームの印刷設定はフォームのプロパティとなり、個別に定義されます。

以前のバージョンにおいて、これらのパラメータは「ファイル」メニューから呼び出される標準の「印刷設定」ダイアログボックスで定義され、フォームとともに保存されていました。つまり、フォームの印刷設定はカレントアプリケーションの印刷設定でした。

4th Dimension 2004 では、アプリケーションの印刷設定（「デザイン」モードで使用）とフォームの印刷設定（「ユーザ」ならびに「カスタム」モードで使用）は分けられています。

「プロパティリスト」の「印刷」テーマ（フォームのプロパティで代入）にある「設定/編集」ボタンは、フォーム特定の「印刷設定」ダイアログボックスを表示します。







これらの設定は、そのフォームとともに保存されます。

「ファイル」メニューから定義された印刷設定は、「デザイン」モードのカレントセッションに対して用いられ、保存されません。

ズーム

カレントフォーム内でズーム（拡大表示）を行えるようになりました。

虫めがねのボタン  をクリックするか、目的のバー  を直接クリックして、「ズーム」モードへ移行することができます。表示倍率は、50%、100%、200%、400%、800%です。

- 虫めがねのボタンをクリックすると、カーソルの形が  に変わります。表示倍率を上げるにはフォーム上でクリックします。また、表示倍率を下げるには、Shift キーを押しながら（カーソルの形は  に変わる）フォーム上でクリックします。
- 倍率バー上でクリックすると、表示が即座に変わります。


「ズーム」モードでは、「フォーム」エディタの全機能を利用できます。

ビューの管理

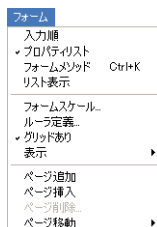
4th Dimension 2004 のフォーム上でビューを使用することができます。この新しい機能により、オブジェクトをさまざまなビューに展開することが可能で、さらにこれらのビューの非表示、表示、ロックを行えるため、複雑なフォームの作成が容易になります。例えば、オブジェクトタイプ（フィールド、変数、スタティックオブジェクトなど）に応じて、オブジェクトを展開することができます。サブフォームおよびプラグインエリアを含め、すべてのタイプのオブジェクトをビューに納めることができます。

フォームごとに9つのビューを利用することができ、デフォルトとしてView 1からView 9までの名前が付けられています。各ビューは、表示や非表示、ロックが可能です。ビューの管理はビューパレットを用いて行います。



このパレットを表示するには、そのウインドウのツールバーにあるビューボタン  をクリックするか、エディタのコンテキストメニューまたは「フォーム」メニューの「表示」サブメニューから、「ビューリスト...」コマンドを選択します。

コンテキストメニュー



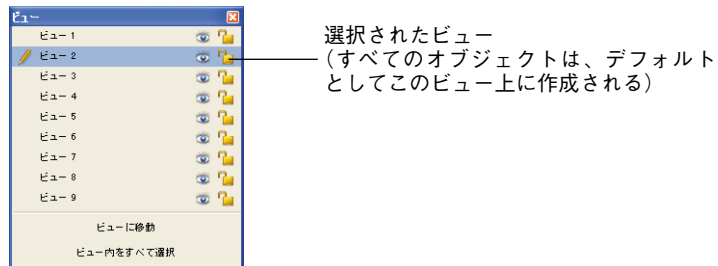
注

- ・ビューは単なるグラフィックツールであり、「フォーム」エディタ上でのみ使用することができます。プログラムや「ユーザ」モードからビューにアクセスすることはできません。
- ・同じビューのオブジェクトが別々のフォームページに属していても構いません。ビューの構成とは関係なく、カレントページ（および、表示されている場合にはページ0）のオブジェクトだけが表示されます。
- ・ビューはオブジェクトの階層からは独立しており、各種ビュー間に表示階層は存在しません。
- ・グループ化することができるのは、カレントビューに属するオブジェクトだけです。

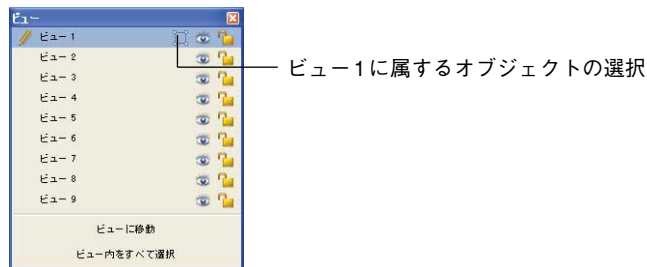
オブジェクトをビューに配置する

あるオブジェクトは、一つのビューにしか属することができません。フォーム上に作成されたすべてのオブジェクトはカレントビューに配置されます。デフォルトでは、ビュー1が選択されています。したがって、オブジェクトはすべてフォームの1番目のビューに配置されます。

オブジェクトを別のビュー上に作成するには、まずパレット上でそのビューの行をクリックして選択します。

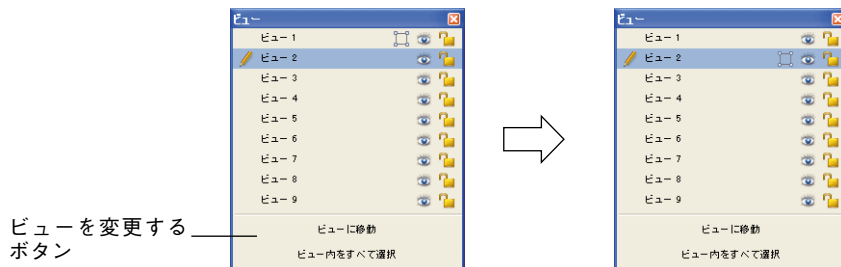


また、あるビューから別のビューへオブジェクトを移動することもできます。これを行うには、ビューを変更したいオブジェクトをフォームから選択します。ビューリスト上ではシンボルを使用して、選択項目が属すビューが示されます。



注：異なるビューに属しているオブジェクトを複数、選択項目の中にも含めることができます。

次に、選択項目の移動先となるビューを選び、「ビューに移動」をクリックします。すると、選択項目は新しいビューに配置されます。



注：4th Dimensionのバッジを使用して、各オブジェクトが含まれているビューを表示することができます。この件に関する詳細は、後述の「バッジを使用する」の節を参照してください。

ビュー名の変更

例えば、オブジェクト内容を表わす名前を指定したい場合に、9つのビューの名前をそれぞれ変更することができます。データベースの各フォームにおいて、9つのビューの名前をそれぞれ変えることができます。

ビュー名を変更するには、ビュー名の上でCtrl+クリック（Windows）またはCommand+クリック（Mac OS）するか、ビュー名をダブルクリックします（この場合は選択したビュー）。すると、名前は編集可能になります。



ビューを使用して作業する

ビューに各オブジェクトを配置したら、ビューパレットを用いて次のような事柄を実行できます。

- 一度のクリックで同一ビュー上のオブジェクトをすべて選択する。
- 各ビューのオブジェクトを表示、または非表示にする。
- ビューのオブジェクトをロックする。
- ビューの全オブジェクトを選択する

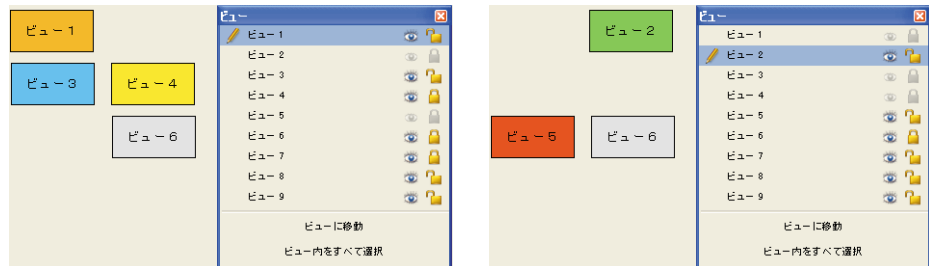
フォームのカレントページにおいて、同一ビューに属するすべてのオブジェクトを選択することができます。一連のオブジェクトを一括変換する場合には、この機能が役立ちます。

これを行うには、全オブジェクトを選択しようとするビューを選び、「ビュー内のすべてを選択」をクリックします。

■ ビューのオブジェクトの表示／非表示

フォームのカレントページにおいて、いつでもビューのオブジェクトを表示したり、隠すことができます。例えば、この機能を使用して、フォームの編集中に特定のオブジェクトの作業に集中することができます。

デフォルトでは、ビューパレット上の各ビューの アイコンで示されるように、すべてのビューが表示されています。あるビューを隠すには、このアイコンをクリックします。するとアイコンがグレー表示され、対応するビューのオブジェクトはフォーム上に表示されなくなります。




注：カレントビュー（ビューリスト上で選択されているビュー）を隠すことはできません。

隠されたビューを表示するには、そのビューを選択するか、表示アイコンを再度クリックします。

■ ビューのオブジェクトをロックする

ビューのオブジェクトをロックすることができます。これにより、オブジェクトの選択や変更、またはフォームからの削除ができなくなります。一度ロックを行うと、クリックや矩形、またはコンテキストメニューの「同種のオブジェクトを選択」コマンドにより、オブジェクトが選択できなくなります。この機能は、操作上のエラー防止に役立ちます。

デフォルトでは、ビューパレットの各ビューの  アイコンで示されるように、すべてのビューはロックされていません。

ビューのオブジェクトをロックするには、前述のアイコンをクリックします。すると南京錠が閉じられ、そのビューがロックされたことを表わします。



注：カレントビュー（ビューリスト上で選択されているビュー）をロックすることはできません。

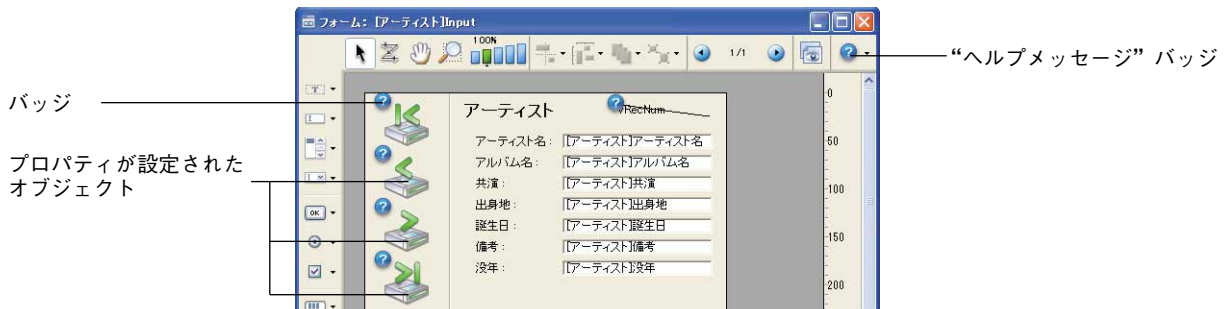
ビューのロックを解除するには、そのビューを選択するか、ロックアイコンを再度クリックします。

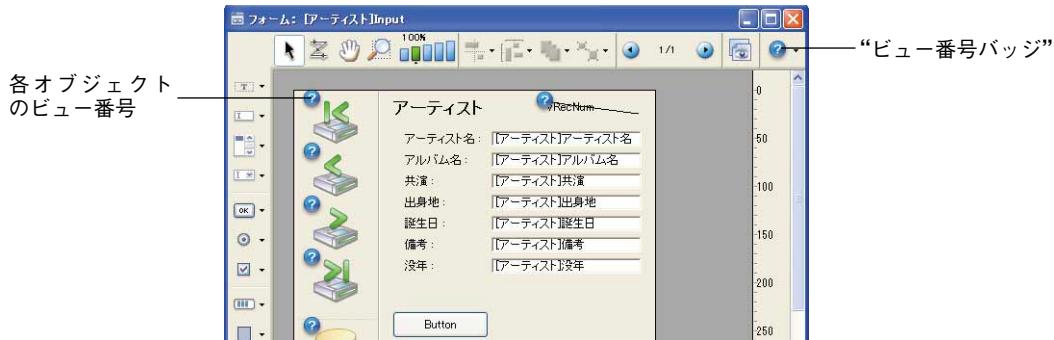
バッジを使用する

4th Dimension 2004の「フォーム」エディタでは、バッジを使用してオブジェクトプロパティをより簡単に表示することができます。

この機能は非常にシンプルです。つまり、各バッジは1つのプロパティに関連しています（例えば、ヒント、つまり関連したヒントが存在することを表わす）。バッジをアクティブにすると、4th Dimensionはプロパティが適用されている各フォームオブジェクトの左上に小さなアイコンを表示します。

“True（真）／False（偽）” タイプのプロパティに関しては（ヒントなど）、オブジェクトにプロパティが設定されている場合にのみ、バッジが表示されます。“値” タイプのプロパティの場合（“ビュー番号” など）、バッジには各オブジェクトのプロパティの値が表示されます。








バッジをアクティブにするには、目的のバッジが選択されるまでバッジ選択ボタンをクリックします。また、このボタンの右側をクリックして、関連メニューから表示するバッジタイプを直接選択することもできます。

















バッジを表示したくない場合、選択メニューから「バッジなし」を選択します。

注：アプリケーションの「環境設定」の「フォームエディタ」ページ（「デザインモード」テーマ）において、デフォルトとして表示するバッジを設定することができます。詳細については、前述の「フォームエディタ」の節を参照してください。

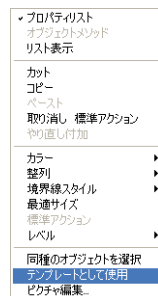
次に、各バッジのタイプを説明します。

アイコン	名称	説明
	オブジェクトメソッド	オブジェクトメソッドが関連付けられたオブジェクトに対して表示される。
	自動アクション	自動アクションが関連付けられたオブジェクトに対して表示される。
	リサイズ	少なくとも1つのリサイズプロパティが設定されたオブジェクトに対して表示され、現在のプロパティの組み合わせを表わす。

	水平方向拡大	
	水平方向移動	
	垂直方向拡大	
	垂直方向移動	
	水平方向拡大/垂直方向拡大	
	水平方向拡大/垂直方向移動	
	水平方向移動/垂直方向拡大	
	水平方向移動/垂直方向移動	
	入力順序	入力可オブジェクトに対して表示され、入力順序を示す。
	ビュー番号	すべてのオブジェクトに対して表示され、ビュー番号を示す。
	スタイルシート	スタイルシートが関連付けられたオブジェクトに対して表示される。
	フィルタ	入力フィルタが関連付けられた入力可オブジェクトに対して表示される。
	ヒント	ヒントが関連付けられたオブジェクトに対して表示される。
	バッジなし	バッジは表示されない。

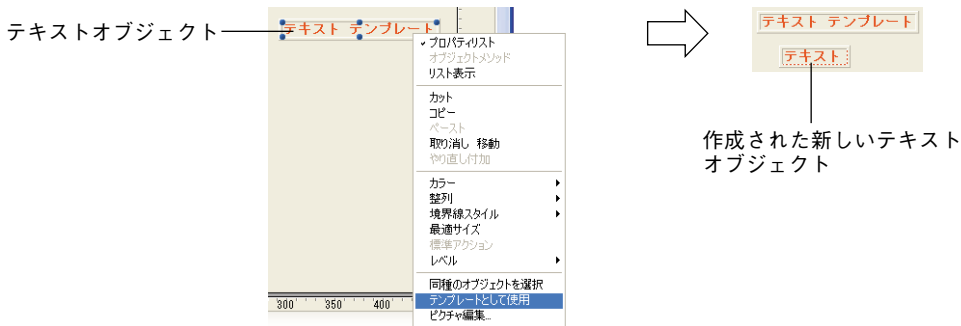
テンプレートとして使用

オブジェクト上でクリックすると表示される「フォーム」エディタのコンテキストメニューにおいて、「テンプレートとして使用」という新しいコマンドが利用できるようになりました。



このコマンドは、オブジェクト（クリックされたオブジェクト）をテンプレートとして保存します。テンプレートとそのプロパティは、同じタイプのオブジェクトを作成する時に利用されます。

次の例では、このコマンドがテキストオブジェクトに適用されています。この後に作成されるテキストオブジェクトはすべて、設定されたオブジェクトのプロパティをデフォルトとして使用します。



各オブジェクトタイプに対し、カスタマイズしたテンプレートを使用することができます。

データベース全体を通して、オブジェクトタイプごとに1つのカスタマイズテンプレートがあります。

固定座標とオブジェクトメソッド（該当する場合）を除き、コマンドの実行時に設定されていたすべてのオブジェクトプロパティがテンプレートに保存されます。

オブジェクトテンプレートを削除するには、新しいテンプレートを作成しなくてはなりません。また、アプリケーションの「環境設定」の「フォームエディタ」ページ（「デザインモード」テーマ）において「カスタムテンプレートをすべてクリアする」ボタンを使用し、オブジェクトテンプレートをすべて削除することもできます（標準のテンプレートに戻る）。詳細については、前述の「フォームエディタ」の節を参照してください。

グループ化されたオブジェクトや背景オブジェクトを選択する

4th Dimension 2004では、ショートカットキーであるCtrl+クリック（Windows）やCommand+クリック（Mac OS）の動作が変更されています。このショートカットを使用すると、グループ化されたオブジェクトや背景オブジェクトを簡単に選択できるようになります。

■ グループ化されたオブジェクト：Ctrl+クリックまたはCommand+クリックを使用すると、グループ化されたままの状態、グループに属するオブジェクトを選択することができます（事前にグループ解除する必要なし）。

- 背景オブジェクト：さまざまなオブジェクトがいくつかのレベルで重なり合っている場合、Ctrl+クリックまたはCommand+クリックを使用して、各オブジェクトを連続して選ぶことができます。この場合、クリックするたびに1つ下のレベルに進みます。

このショートカットの従来の動作は（同じタイプのオブジェクトを選択）、エディタのコンテキストメニューの「同種のオブジェクトを選択」アイテムを使用して実行することができます。

マグネティックグリッドを使用する

4th Dimension 2004では、“マグネティック”オブジェクト整列機能が変更されています。

マグネティックオブジェクト整列機能は、その間にあるオブジェクトの相対位置に基づいて適用されます。マグネティック機能は、フォーム上に少なくとも2つのオブジェクトが存在する場合にのみ利用できます。「フォーム」エディタ上で、グリッドを常時表示することは不可能になります。

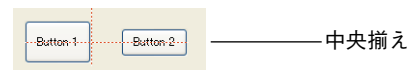
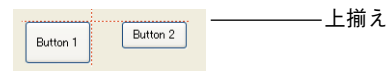
マグネティックグリッドの機能は次の通りです。

フォームのオブジェクトをドラッグ&ドロップする際、4th Dimensionは他のフォームオブジェクトとの相対的な配置に基づいて、このオブジェクトの位置候補を示します。

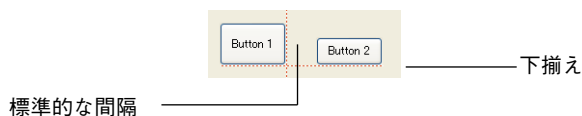
- 2つのオブジェクトの端または中央が一致する場合、水平方向の位置候補が提示されます。

- 2つのオブジェクトの端が一致する場合、垂直方向の位置候補が提示されます。

この時点で、4th Dimensionはその位置にオブジェクトを配置し、考慮された整列を視覚的に示す赤いラインを表示します。



オブジェクトの配置に関し、4th Dimensionはインターフェース標準に基づいた間隔を提示します。マグネティック整列と同様に、決定した時点でその差分を赤いラインで視覚的に示します。




最後に、マグネティックグリッドは手動でサイズ変更を行うオブジェクトにも影響を及ぼします。

これらの新しいメカニズムにより、機能的なインタフェースを素早く作成することができます。もちろん、これはあらゆるタイプのフォームオブジェクトに適用されます。以前のバージョンの4th Dimensionのように、「フォーム」メニューの「マグネティックグリッドを使用する」コマンド、またはエディタのコンテキストメニューを使用し、いつでもマグネティック機能を使用可または使用不可に設定することができます。また、アプリケーションの「環境設定」の「フォームエディタ」ページ（「デザインモード」テーマ）において、この機能をデフォルトとしてアクティブに設定しておくことも可能です。詳細については、前述の「フォームエディタ」の節を参照してください。

注：Ctrlキー（Windows）またはCommandキー（Mac OS）を使用してオブジェクトを選択すると、マグネティックグリッドを手動で有効／無効に設定することができます。

ユーザ編集可

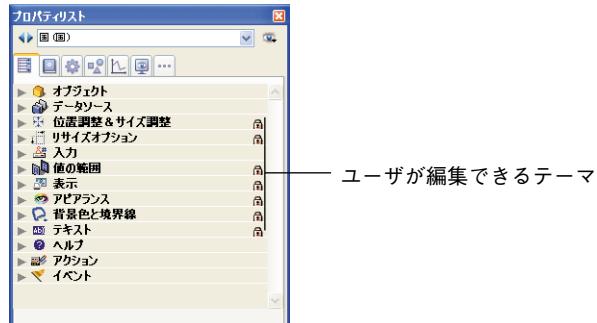
新しい「ユーザ編集可」プロパティにより、そのフォームが「ユーザフォーム」エディタ上で表示可能であることを示すことができます（後述の「ユーザフォーム」の節を参照）。

 このプロパティを選択すると、保存時にフォームが自動的にロックされます。次回、そのフォームをロック解除せずに開くと、選択以外の動作は許可されなくなります（警告音が鳴る）。このフォームを修正するには、南京錠の形のアイコンをクリックし、フォームのロックを明示的に解除しなければなりません。フォームのロックを解除すると、その他の定義されたユーザフォームが陳腐化するため注意してください。このメカニズムに関する詳細は、後述する「ユーザフォーム」の節を参照してください。

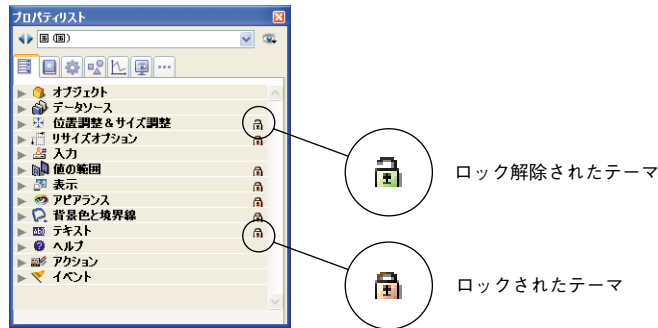
「ユーザ編集可」のフォームのロックを解除すると、「ユーザフォーム」エディタ上で編集可能となるフォームや各オブジェクトのプロパティを設定することができます。設定は、「プロパティリスト」の各プロパティテーマで行います。

編集可として宣言できるのは特定のプロパティだけです。特に、プログラミングに関連するプロパティ（メソッド、イベントなど）はすべて編集可にすることができません。

「プロパティリスト」において、編集可能であるプロパティは南京錠アイコンで示されません。デフォルトでは、これらのプロパティがすべてロックされています（すべてのプロパティはユーザ編集不可）。



プロパティテーマのロックを解除するには（これにより、「ユーザフォーム」エディタで編集可能）、南京錠アイコンをクリックします。同じ操作を行うと、再びロックすることができます。



ダイナミック調整

「ダイナミック調整」フォームプロパティを使用して、フォームのサイズを変更することができます。以前のバージョンの4th Dimensionから変換されたフォームで「自動サイズ」プロパティが選択されている場合に、フォームの「フォームサイズ」テーマにおいてこのプロパティを利用することができます。

以前のバージョンの4th Dimensionでは、フォームに「自動サイズ」プロパティが設定されていると、そのフォームが開かれた時にだけフォームサイズが計算されていました。この後、**MOVE OBJECT**のようなコマンドを使用して変更を行っても、フォームサイズは調整されませんでした。

4th Dimension 2004では、この場合でもフォームサイズが動的に調整されます。

この原則は、バージョン2004で作成されたフォームに対して有効です。その反面、互換性上の理由から変換後のデータベースのフォームには、このプロパティがデフォルトとして選択されていません。変換後のフォームのサイズを動的に調整したい場合は「ダイナミック調整」オプションを選択してください。

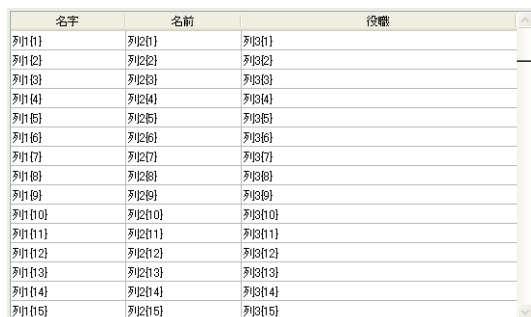
フォームオブジェクト

4th Dimension 2004のフォームオブジェクトには、数々の新機能が追加されています。

- 新しいオブジェクトタイプ、リストボックスの導入。
- スプリット処理の向上
- プラットフォームインタフェース管理の新しい概念
- ネイティブな入力制御の使用
- フィールドと変数のスペルチェック
- 3ステートチェックボックス
- 新しい3Dボタン
- 最後に、プルタイプのチェックボックス、組み込み参照の変更、新しい「デフォルトで非表示」プロパティ、グループ化されたラジオボタンなどの各種機能が拡張、または改良されています。

リストボックス

4th Dimension 2004では、リストボックスという新しいタイプのフォームオブジェクトが提供されています。



名字	名前	役職
列名1	列名1	列名1
列名2	列名2	列名2
列名3	列名3	列名3
列名4	列名4	列名4
列名5	列名5	列名5
列名6	列名6	列名6
列名7	列名7	列名7
列名8	列名8	列名8
列名9	列名9	列名9
列名10	列名10	列名10
列名11	列名11	列名11
列名12	列名12	列名12
列名13	列名13	列名13
列名14	列名14	列名14
列名15	列名15	列名15

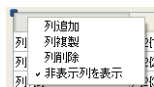
リストボックス

リストボックスは、同期化されたカラムとしてデータの表示や入力を行えるエリアです。これらのオブジェクトは、以前のバージョンの4th Dimensionにおける“グループ化したスクロールエリア”と同じですが、各種機能は削除、または拡張されています（値入力、カラムの並び替え、外観のカスタマイズ、ドラッグ&ドロップなど）。リストボックスタイプのオブジェクトはすべて、4th Dimensionの「フォーム」エディタで設定可能であり、また、プログラムからも処理することができます。

この節では、「フォーム」エディタにおけるリストボックスタイプのオブジェクトの作成および設定に関する新機能について説明します。プログラム上でこれらオブジェクトを管理する上での情報については、後述の「リストボックス」の節を参照してください。

主な特徴

リストボックスは1つ以上のカラムを含み、それらの内容は自動的に同期化されます。デフォルトとして、作成時にリストボックスは1つのカラムを含みます。コンテキストメニュー（カラムヘッダをクリック）を使用するか、リストボックスのプロパティにおいて、カラム数を変更することができます（カラムの追加、コピー、削除）。

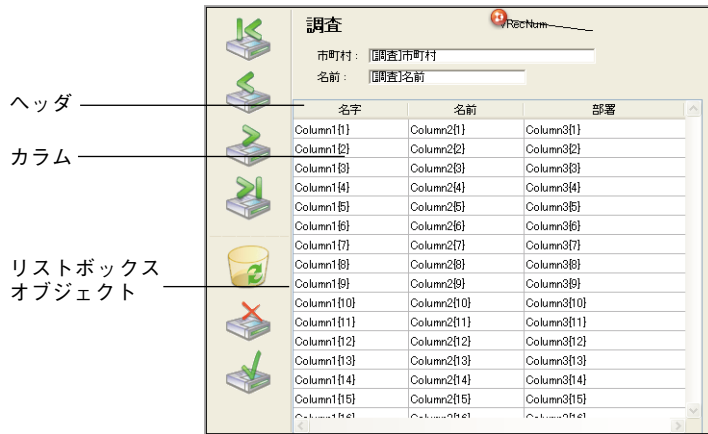


理論上、カラム数に制限はありません（マシン資源に依存）。

リストボックスの各カラムは、4D配列に関連付けられています。デフォルトとして、4th Dimensionはカラムの各変数、つまり関連する各配列に対し“ColumnX”という名前を割り当てます。この名前はカラムのプロパティで変更することができます。リストボックスに入力、または表示された値の管理は、4th Dimensionランゲージを用いて行います。この件に関する詳細は、後述の「“リストボックス”オブジェクトタイプのプログラミング入門」の節を参照してください。入力を管理するために、リストをカラムに関連付けることができます（後述の「リストボックス固有のプロパティ」の節を参照）。

リストボックスは、オブジェクト全体、カラム、カラムヘッダという異なる3つの部分から成り立ちます。「フォーム」エディタで、これらの部分を別々に選択することもできます。

各部分には、それぞれ独自のオブジェクトと変数名、ならびに固有のプロパティがあります。



例えば、カラム数や各行の代替カラーはリストボックスのオブジェクトプロパティで定義します。各カラムの幅は、カラムプロパティで指定し、ヘッダのフォントはヘッダプロパティで定義します。

「ユーザ」モード

「ユーザ」モードでは、リストボックスを使用してデータをリストとして表示、または入力することができます。

セルを入力可にするには（そのカラムの入力が許可されている場合）、セルに納められている値をダブルクリックします。

名字	名前
鈴木	次郎
柳田	洋一

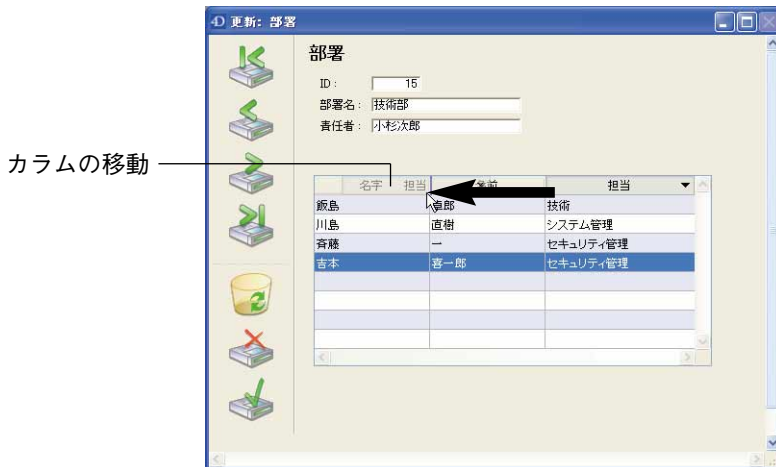
また、ヘッダをクリックしてカラムの値を並び替えることができます。複数回クリックすることにより、並び替えはアルファベット順、降順、昇順と交互に行われます。

すべてのカラムは自動的に同期化されます。

各カラムのサイズを変更することも可能です。



ユーザはマウスを用いてカラムや行を移動し、その順序を変更することができます（この動作が許可されている場合）。



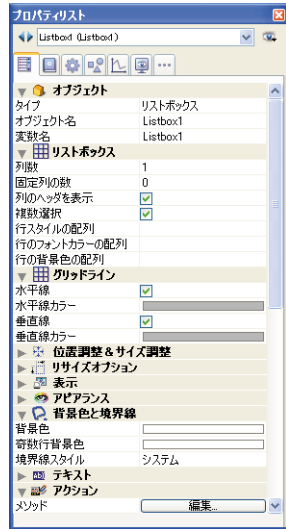
最後に、標準のショートカットを使い、1つ以上の行を選択することができます。連続する行の選択にはShift+クリック、連続していない行の選択にはCtrl+クリック（Windows）またはCommand+クリック（Mac OS）を使用します。



これらの機能はすべて、リストボックス、カラム、ヘッダのプロパティを用いて処理することができます。この件については、次の節で詳しく説明します。

リストボックス固有のプロパティ

「フォーム」エディタ上でリストボックスオブジェクトを選択すると、次のようなプロパティの一覧が表示されます。

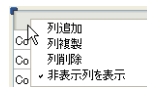


この節では、リストボックスオブジェクトに固有のプロパティについて説明します。

■「リストボックス」テーマ

このテーマには、基本的なオブジェクトプロパティがすべて集められています。

■ **列の数**：リストボックスに表示されるカラム数（デフォルトは1）。コンテキストメニュー（カラムやカラムヘッダをクリック）を使用して、カラム数を変更することができます（カラムの追加、コピー、削除）。



■ **固定列の数**：「ユーザ/カスタム」モードで移動できないカラムの数。この値は、リストボックスの一番目のカラムから数えた固定カラムの数を表わします。あらゆるカラムの移動を禁止するには、この値とカラムの合計数とを同じにしなくてはなりません。このプロパティは、非表示カラムも計算に入れる点に注意してください。

このプロパティのデフォルト値は0であり、これはすべてのカラムが移動可能であることを意味します。

新しい **MOVED LISTBOX COLUMN NUMBER** コマンドを使用すると、リストボックスカラムの移動を管理することができます。

- **列のヘッダを表示**：カラムヘッダの表示／非表示を指定します（デフォルトは表示）。

- **複数選択**：複数行の選択を許可、または禁止します。デフォルトでは、標準のショートカットを用いて複数行を選択することができます。連続した行の選択には **Shift** キー＋クリック、連続していない行の選択には **Ctrl** キー＋クリック（Windows）または **Command** キー＋クリックを使用します。

このオプションをチェックしない場合、リストボックスの行は一行ずつしか選択できません。

- **行スタイル配列**：リストボックスの各行に対して独自の文字スタイルを割り当てる際に使用する、倍長整数配列の名前を入力することができます。この配列の各要素は、リストボックスの行に対応します。つまり、この配列のサイズはカラムに関連付けた配列のサイズと同じでなくてはなりません。配列へ値を代入するには（メソッドを使用）、「**フォントスタイル**」テーマの定数を使用します。定数を加えることにより、スタイルを組み合わせることができます。リストボックスのプロパティで定義されたスタイルを行に適用するには、対応する配列の要素に対して **-255** という値を渡します。

- **行フォントカラー配列**：リストボックスの各行に対して独自の文字フォントカラーを割り当てる際に使用する、倍長整数配列の名前を入力することができます。この配列の各要素は、リストボックスの行に相当します。つまり、この配列のサイズはカラムに関連付けた配列のサイズと同じでなくてはなりません。

この配列には **RGB** カラーの値を代入しなくてはなりません（メソッドを使用）。この件に関する詳細は、**SET RGB COLORS** コマンドを参照してください。

リストボックスのプロパティで定義されたフォントカラーを行に適用するには、対応する配列の要素に対して **-255** という値を渡します。

- **行背景色配列**：リストボックスの各行に対して独自の背景色を割り当てる際に使用する、倍長整数配列の名前を入力することができます。この配列の各要素は、リストボックスの行に相当します。つまり、この配列のサイズはカラムに関連付けた配列のサイズと同じでなくてはなりません。

この配列には **RGB** カラーの値を代入しなくてはなりません（メソッドを使用）。この件に関する詳細は、**SET RGB COLORS** コマンドを参照してください。

リストボックスのプロパティで定義された背景色を行に適用するには、対応する配列の要素に対して **-255** という値を渡します。

注：行スタイル、行フォントカラー、行背景の各配列は、**LIRE TABLEAUX LISTBOX** コマンドにより返されます。

■「グリッドライン」テーマ

このテーマには、リストボックスオブジェクトに表示されるグリッドに関連するプロパティがすべて集められています。

■ **横線**：リストボックスの水平ラインの表示や非表示を設定します（デフォルトは表示）。

■ **横線色**：リストボックスの水平ラインのカラーを設定します（デフォルトはグレー）。

■ **縦線**：リストボックスの垂直ラインの表示や非表示を設定します（デフォルトは表示）。

■ **縦線色**：リストボックスの垂直ラインのカラーを設定します（デフォルトはグレー）。

■「位置調整&サイズ調整」テーマ

このテーマには、リストボックスの座標や幅、高さに関連するプロパティがすべて集められています。

「行の高さ」プロパティは固有のもので、このプロパティを使用して、リストボックスの行の高さを指定することができます（ピクセル単位）。この高さはヘッダに対しても適用されます。デフォルトとして、行の高さはプラットフォームやフォントサイズに応じて定義されます。

■「リサイズオプション」テーマ

このテーマには、フォームサイズが変更された場合のリストボックスの対応に関する標準的なプロパティがすべて集められています。

■「表示」テーマ

このテーマには、フォームにおけるリストボックスの表示ステータスに関する標準的なプロパティがすべて集められています。「デフォルトで非表示」プロパティに関する詳細は、後述の「新しいデフォルトで非表示プロパティ」の節を参照してください。

■「アピアランス」テーマ

このテーマには、リストボックスのプラットフォームのインターフェースに関連する標準的なプロパティがすべて集められています。必要があれば、横スクロールバーや縦スクロールバーを使用するかどうかをこのテーマで指定することもできます。

■背景と境界線

このテーマには、行の背景色ならびに境界線スタイルに関するプロパティが再編成されています。

■ **背景色**：リストボックスの背景色を設定します。このカラーは、ヘッダ（表示される場合）を除いた全オブジェクトに使用されます。

- **代替背景色**：リストボックスの奇数行に使用される別の背景色を設定します。代替背景色を利用すると、配列がより見やすくなります。
- **境界線スタイル**：リストボックスオブジェクトの境界線に対する標準スタイルを定義します。

■ 「テキスト」 テーマ

このテーマには、リストボックスに表示されるテキストに関する標準的なプロパティがすべて集められています（スタイルシート、フォント、属性など）。各カラムおよびヘッダに対し、特定のテキストプロパティを指定できる点に注意してください。

■ 「アクション」 テーマ

このテーマには、リストボックスのダイナミック動作に関するプロパティがすべて集められています。

- **メソッド（編集...）**：このボタンはリストボックスオブジェクトのメソッドを表示します（カラムやヘッダにはオブジェクトメソッドを指定できない点に注意）。
- **移動可能行**：「ユーザ」または「カスタム」モードにおいて行の移動を許可します。デフォルトとして、このオプションが選択されています。新しい **MOVED LISTBOX ROW NUMBER** コマンドを使用し、これらリストボックス行の移動を管理することができます。

- **並び替え可**：ヘッダクリックによるデータの並び替えを許可します。デフォルトでは、このオプションが選択されています。

この方法で、ピクチャタイプの配列（カラム）を並び替えることはできません。

■ 「イベント」 テーマ

このテーマには、リストボックスに関連するフォームイベントがすべて集められています。4th Dimension 2004では、リストボックスプログラミングの仕様を処理するために、新たなイベントが追加されています。詳細については、後述の「リストボックスに関連するフォームイベント」の節を参照してください。

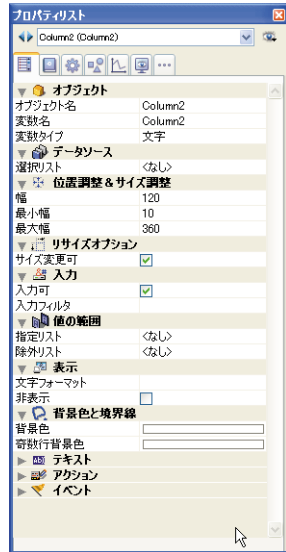
リストボックスカラム固有のプロパティ

「フォーム」エディタ上でリストボックスオブジェクトが選択されている場合、カラムをクリックしてリストボックスカラムを選択することができます。

名子	名前	部署
Column1[1]	Column2[1]	Column3[1]
Column1[2]	Column2[2]	Column3[2]
Column1[3]	Column2[3]	Column3[3]
Column1[4]	Column2[4]	Column3[4]
Column1[5]	Column2[5]	Column3[5]
Column1[6]	Column2[6]	Column3[6]
Column1[7]	Column2[7]	Column3[7]

選択されたカラム

この場合、次のような「プロパティリスト」が表示されます。

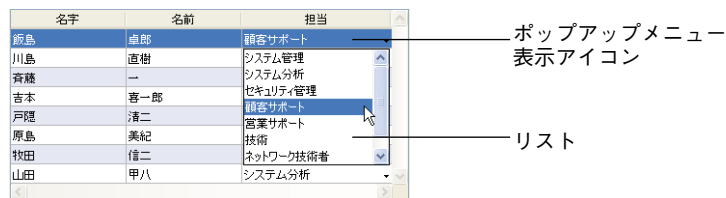


この節では、リストボックスカラムに固有のプロパティについて説明します。

■「データソース」テーマ

このテーマ内のプロパティを使用し、「選択リスト」をリストボックスカラムに関連付けることができます。

リストを設定すると、ユーザは独自の値（ポップアップメニューで表示）を用いて、カラムの値や関連する配列を修正することができます。



キーボード入力が今まで通り行える点に注意してください。ポップアップ選択メニューを使用して値の変更だけを行いたい場合、必要となるリストを選択してください（後述）。

注：指定したリストが階層構造である場合、第一階層の要素だけが考慮されます。

■「位置調整&サイズ調整」テーマ

このテーマには、リストボックスのカラム幅に関連するプロパティが集められています。

- **幅**：デフォルトのカラム幅（ピクセル単位）。「フォーム」エディタ上でマウスを使用してカラムサイズを変えると、この値が変更されます。

名	前	リサイズカーソル
Column1[1]	Column2[1]	
Column1[2]	Column2[2]	

「サイズ変更可」プロパティが選択されている場合、手動でカラムサイズを変更することもできます。

- **最小幅**：カラムの最小幅（ピクセル単位）。カラムやフォームのサイズを変更する際、カラム幅はこの値よりも小さくできません。
- **最大幅**：カラムの最大幅（ピクセル単位）。カラムやフォームのサイズを変更する際、カラム幅はこの値よりも大きくできません。

注：フォームサイズを変更する際、リストボックスに「拡大水平サイジング」プロパティが指定されていると、右端に位置するカラムは必要に応じてその最大幅よりも大きくなります。

■「リサイズオプション」テーマ

このテーマは「サイズ変更可」オプション（デフォルトでは選択）だけを含みます。このオプションを選択すると、ヘッダエリアの各辺を動かして、カラムサイズを変更することができます。

■「入力」テーマ

このテーマには、リストボックスカラムのデータ入力に関するプロパティがすべて集められています。

- **入力可**：カラム入力を許可します（デフォルトでは選択）。セルの値を変更するには、その値をダブルクリックしなくてはなりません。

このプロパティを無効にすると、リストを通してカラムに関連付けられたポップアップメニュー（“データソース”と“値の範囲”テーマ）は無効になります。

- **入力フィルタ**：カラムのセルに入力フィルタを関連付けます。「入力可」プロパティが選択解除されている場合、このプロパティは利用できません。

■「値の範囲」テーマ

このテーマのプロパティを使用して、リストボックスカラムの入力管理に用いるリストを設定することができます。

- **指定リスト**：カラムに単一の値だけを挿入できるリストを設定することができます。リストの値は、各セルに関連付けたポップアップメニューを用いて取得することができます（「データソース」テーマの「選択リスト」プロパティを参照）。

「選択リスト」プロパティとは異なり、指定リストを定義するとキーボード入力是不可能となり、ポップアップメニューによるリストの値の選択だけが許可されます。「選択リスト」プロパティと「指定リスト」プロパティを使用して別々のリストが定義されている場合は、「指定リスト」プロパティが優先されます。

- 除外リスト：カラムに入力できない値のリストを設定することができます。除外対象の値を入力すると、その値は受け入れられず、エラーメッセージが表示されます。

注：指定したリストが階層構造である場合、第一階層の要素だけが考慮されます。

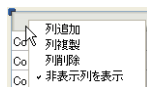
■「表示」テーマ

このテーマを使用し、カラムの値の表示フォーマットや非表示プロパティを指定することができます。

- 「フォーマットタイプ」メニュー上で利用できるプロパティは、「オブジェクト」テーマで定義した変数のタイプに応じて自動的に更新されます。4th Dimensionで使用可能な標準のフォーマットは、文字、数値、日付、時間、ピクチャ、ブルです。テキストタイプに特定の表示フォーマットはありません。既存のカスタマイズフォーマットがあれば、それも使用することができます。

注：ブルタイプの配列は、チェックボックスまたはポップアップメニューとして表示することができます。「チェックボックス」オプションを選択した場合、「タイトル」プロパティを用いてチェックボックスのラベルを入力することができます。「ポップアップメニュー」オプションを選択した場合は、「テキスト (True時)」および「テキスト (False時)」プロパティが表示され、ポップアップメニューに対しこれら2つのラベルを設定することができます。

- 「非表示」プロパティが選択されている場合は、「ユーザ」や「カスタム」モードにおいてカラムを隠すことができます。「デザイン」モードでは、リストボックスのコンテキストメニュー（カラムまたはカラムヘッダをクリック）を使用して、非表示のカラムを表示するかどうかを選択できます。



■「背景と境界線」テーマ

このテーマには、カラムの行の独自の背景色に関するプロパティが集められています。デフォルトでは、一般的なリストボックスの背景色が使用されます。

- 背景色：カラムに対して特定の背景色を設定することができます。
- 代替背景色：カラムの奇数行に対し、特定の代替背景色を設定することができます。

■「テキスト」テーマ

このテーマには、カラムに表示されるテキストに関する特定のプロパティが集められています（スタイルシート、フォント、属性など）。デフォルトでは、一般的なリストボックスのテキストプロパティが使用されます。

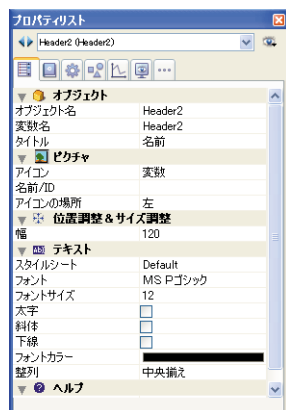
リストボックスヘッダ固有のプロパティ

リストボックスオブジェクトが選択されている場合に、「フォーム」エディタ上でリストボックスのヘッダをクリックしてそれを選択することができます。

選択されたヘッダ

名字	名前	担当
Column1[1]	Column2[1]	Column3[1]
Column1[2]	Column2[2]	Column3[2]
Column1[3]	Column2[3]	Column3[3]
Column1[4]	Column2[4]	Column3[4]

この場合、次のような「プロパティリスト」が表示されます。



この節では、リストボックスのヘッダに固有のプロパティについて説明します。

■「オブジェクト」テーマ

このテーマには、ヘッダ定義に使用するプロパティが含まれます。

■ オブジェクト名：ヘッダオブジェクトの名前

■ **変数名**：ヘッダオブジェクトに関連付けられた変数の名前。この変数（数値）を使用し、カラムの現在の並び替えや、矢印による並び替えの表示をプログラムから管理することができます（後述の「“リストボックス” オブジェクトタイプのプログラミング入門」の節を参照）。

■ **タイトル**：ヘッダに表示するラベル

■「ピクチャ」テーマ

このテーマには、カラムヘッダにピクチャを表示するプロパティが含まれています (任意)。特に、カスタマイズした並び替えを実行する場合に、カラムタイトルの横、あるいはカラムタイトルの代わりとしてアイコンをヘッダに表示することができます。

名前	担当
卓郎	技術

- **アイコン**：ヘッダに挿入するピクチャの参照先を設定します。4th Dimensionのピクチャボタンと同様に、変数やピクチャライブラリ、リソースファイルのピクチャを使用できます。
- **名前/ID**：事前に定義した参照先で使用するピクチャを設定します。このフィールドに入力する情報は、選択した参照先に応じて変わります。
- **アイコンの位置**：ヘッダ上のアイコンの位置。ヘッダの左側、または右側にアイコンを配置することができます。

■「位置調整&サイズ調整」テーマ

このテーマには「幅」プロパティが含まれ、その省略値はカラムに指定した値と同じです。

■「テキスト」テーマ

このテーマには、ヘッダに表示されるテキストに関連する特定のプロパティが集められています (スタイルシート、フォント、属性など)。デフォルトでは、一般的なリストボックスのテキストプロパティが使用されます。

■「ヘルプ」テーマ

このテーマには「ヘルプヒント」プロパティが含まれ、このプロパティを使用してヘッダのヒントを関連付けることができます。ヒントを設定すると、ユーザがヘッダ上にマウスカーソルを置くとヒントの内容が表示されます。

注：ヒントの定義に関する詳細は、前述の「ヘルプヒント」の節を参照してください。

スプリッタの管理

4th Dimension 2004ではスプリッタ管理が変更されており、スプリッタはより使いやすく、インタフェース標準に準拠するようになります。これらの変更は、次の事柄に関して行われています。

- 新しい「プッシャー (押し込み)」プロパティ
- “反応領域” 管理
- ページ0に表示されるスプリッタの管理
- 「On click」フォームイベントのリアルタイム表示と呼び出し

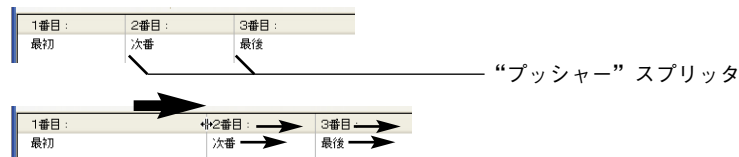
注：ウインドウの各辺も、プッシャースプリッタとして動作します。この件に関する詳細は、後述の「側面プッシャー」の節を参照してください。

プッシャープロパティ

プロパティリストの「リサイズオプション」テーマでは、スプリッタオブジェクトに対して新しいプロパティ「プッシャー」が利用できるようになりました。

スプリッタオブジェクトにこのプロパティを使用すると、そのオブジェクトの右側（縦スプリッタ）または下側（横スプリッタ）にある他のオブジェクトは、スプリッタオブジェクトと一緒に移動します。この時、他のオブジェクトは“プッシュ”または“プル”されます。

“プッシャー” スプリッタを移動した場合の結果を次に示します。



スプリッタに対してこのプロパティが適用されていない場合、結果は次のようになります。



新規データベースには、このプロパティがデフォルトとして選択されていますが、変換されたデータベースのフォームでは無効となります。

反応領域

スプリッタの“反応領域”の管理（ユーザがクリックしてスプリッタをアクティブにできるエリア）、特に、幅が縮小された（例えば1ピクセル）スプリッタオブジェクトの管理が最適化されています。

ページ0のスプリッタ

フォームのページ0に配置されたスプリッタは、すべてのページでアクティブになります。ページ0のスプリッタをそのフォームのページXへ移動すると、他のページにおいて表示の際にその移動結果が考慮されます（例えば、タブを使用）。

リアルタイム表示と「On click」イベント

スプリッタの使用時に、オブジェクトの移動がリアルタイムで表示されるようになります。

以前のバージョンの4th Dimensionとは異なり、移動中は継続して「On click」イベントが生成されるようになります。

プラットフォームインタフェース

4th Dimension 2004では、プラットフォームインタフェース管理がシンプルになりました。ただし、変換されたデータベースでは以前の管理システムが維持されています。

「システム」ならびに「印刷」という2つの新しいプラットフォームインタフェースプロパティが追加されています。フォームインタフェースの管理は、これら2つのプロパティに基づいて行われます。つまり、画面表示の際、フォームは現行のOSのインタフェースを順守してはなりません（システム）。一方、フォームが印刷用に設定されている場合、プラットフォームに関わらずオブジェクトの外観を合わせておかななくてはなりません（印刷）。

システム

この新しいプロパティを使用すると、4th Dimensionが稼働する現行のプラットフォームに合わせて、自動的にフォームやオブジェクトの外観を作り替えることができます。

■ Mac OS X上でフォームが表示される場合、フォームやオブジェクトはMac OS Xのルック&フィールになります。

■ Windows上でフォームが表示される場合は次の通りです。

■ Windows 2000において、フォームやオブジェクトはWindows 2000のルック&フィールになります。

■ Windows XPでは、コントロールパネルの現在の「アピアランス」設定に応じて、フォームやオブジェクトはWindows XPまたはWindows Classicのルック&フィールになります。

注：Windows XP仕様の管理を向上するため、SET RGB COLORSコマンドで新しいカラー定数を利用できるようになりました。

印刷

この新しいプロパティを使用すると、フォームやオブジェクトの外観を印刷用に作り替えることができます。オブジェクトやグラフィック項目（ボタン、チェックボックス、タブ、ラインなど）はすべてベクトルとして描画され、申し分のない印刷結果がもたらされます。

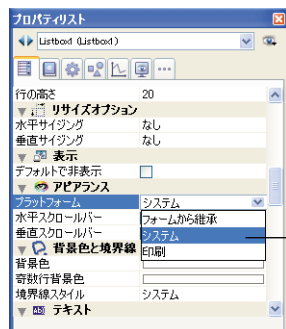
また、フォームを表示または印刷するプラットフォームが何であれ、同じ結果が得られます。

互換性とプロパティへのアクセス

開かれているデータベース（新規または変換済）が作成された条件によって、利用できるプロパティは異なります。

- 新規データベース（4th Dimension 2004以降のバージョンで作成）では、「システム」と「印刷」のプラットフォームプロパティだけが利用できます。これらのプロパティは、以前のバージョンの4th Dimensionで使用可能であった他のすべてのオプションを置き換えます。

データベースのすべてのフォームとオブジェクトに対し、デフォルトとして「システム」オプションが適用されます。これはフォームやオブジェクトレベルでのみ変更できます。

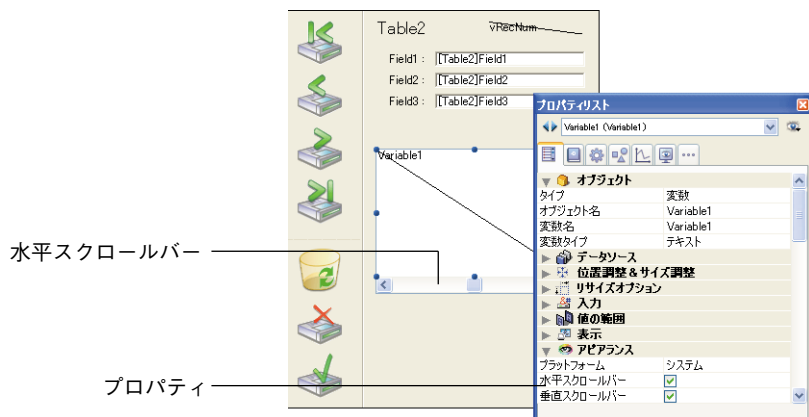


オブジェクトのプラットフォームオプション

最後に、**SET PLATFORM INTERFACE** コマンドや **Get platform interface** 関数はこのプロパティに対して無効であり、使用することはできません。

- 以前のバージョンの4th Dimensionから変換されたデータベースでは、これらのプロパティが他のインタフェースオプションに追加されます。他のインタフェースオプションは互換性上の理由から残されています。以前のバージョンの4th Dimensionと同様に、フォームやオブジェクトの「環境設定」において従来のプラットフォームプロパティを指定することができます。

- 水平スクロールバー（テキストタイプのみ）：ネイティブ制御の各テキストエリアには水平スクロールバーがあります。水平スクロールバーは、オブジェクトプロパティの「アピアランス」テーマで指定します。



- “システム”ドラッグ&ドロップ：ネイティブ制御の各テキストエリアを使用して、“システム”ドラッグ&ドロップを実行することができます。これは、あるエリアから別のエリアへ選択テキストの移動やコピーを行います。この機能は、同じ4Dエリア内や2つの4Dエリア間、あるいは4Dと他のアプリケーション（例えばWordPad）間で利用することができます。ただし、ソースエリアとターゲットエリアがともにネイティブ制御である場合にのみ機能します。



“システム”ドラッグ&ドロップを利用すると、4th Dimensionのドラッグ&ドロップメカニズムが使用されません。したがって、「On drag over」のようなフォームイベントは生成されず、ドロップエリアは「環境設定」で定義されたようにアクティブにはなりません。ネイティブ制御のエリアに対して、4th Dimensionの内部的ドラッグ&ドロップシステムを強制的に使用したい場合は、Altキー（Windows）またはOptionキー（Mac OS）を押下してドラッグ&ドロップを実行します。

- URL検出とアクティブ化：ネイティブ制御のテキストエリアにURL（http、ftp、www、mailtoで開始するテキスト）が含まれる場合は、自動的に検出され、青い下線付きで表示されます。



注：電子メールアドレスは、「mailto:アドレス」という形式で記述しなくてはなりません。

デフォルトでは、そのエリアをクリックすると、カレント Web ブラウザ上でその URL が実行されます。

この標準動作は、2つの新しいプロパティを用いて制御することができます（後述の節を参照）。

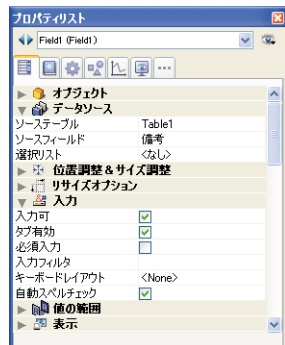
スペルチェック

4th Dimension 2004 には、複数の言語で利用できるスペルチェックユーティリティが統合されています。文字やテキストタイプの変数とフィールド、ならびに 4D Write ドキュメントをチェックすることができます。

スペルチェックのアクティブ化

フィールド、文字、またはテキストタイプの変数に対してスペルチェックをアクティブにする方法は、2通りあります。

- フィールドや変数に対して「スペルチェック入力フィルタ」プロパティ（「入力」テーマ）を選択します。

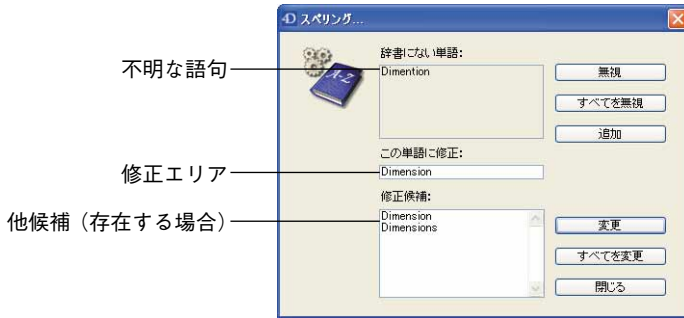


この場合、オブジェクトがフォーカスを失うと、入力中にスペルチェックが自動実行されます。

- 各オブジェクトに対して **SPELL CHECKING** コマンドを実行して制御を行います。このコマンドに関する詳細は、後述の「**SET DICTIONARY**」の節を参照してください。

不明な語句の検出

スペルチェック機能が有効かどうかに関わらず、不明な語句が検出されると、次のダイアログボックスが表示されます。



複数のボタンを利用できます。

- 無視する：不明な語句はそのまま残されます。
- すべて無視する：不明な単語はそのまま残され、このエリアの他の箇所にある同じ語句もすべてそのまま残されます。
- 追加：不明な語句は辞書に追加されます。したがって、今後のスペルチェック機能でその語句は検出されなくなります。
- 変更：不明な語句は、修正エリアに提示された語句で置き換えられます。
- すべて変更：不明な語句、ならびに他の箇所にある同じ語句はすべて、修正エリアに提示された語句で置き換えられます。
- 閉じる：修正は行われず、ダイアログボックスが閉じられます。

注：「閉じる」ボタンを除き、ボタンをクリックするとエリアの最後までスペルチェックが続行されます。

辞書の管理

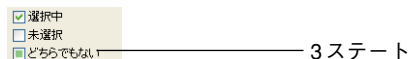
4th Dimension 2004 のスペルチェック機能では、フランス語、英語、ドイツ語、スペイン語の4つの辞書が使用できます。

デフォルトとして、4th Dimension は現行のアプリケーション言語に対応する辞書を使用します。

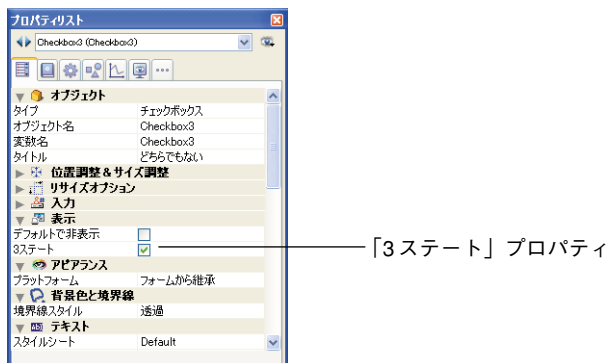
新しい **SET DICTIONARY** コマンドを使用すると、別言語の辞書を強制的にオープンすることができます。

3 ステートチェックボックス

4th Dimension 2004では、チェックボックスオブジェクトで3番目の状態が受け入れられるようになります。この3番目の状態とは中間的な状態であり、通常は表示目的に使用されます。例えば、これにより、あるプロパティがすべてのオブジェクトの中ではなく、選択オブジェクトの中に存在することを表現できます。



チェックボックスでこの3番目の状態を制御できるようにするには、「プロパティリスト」の「表示」テーマにおいて「3ステート」プロパティを選択しなくてはなりません。



このプロパティは、数値タイプの変数に関連付けた標準のチェックボックスに対してのみ使用できます。3Dチェックボックスやブールフィールド用のチェックボックスに対して、この「3ステート」プロパティを利用することはできません（ブールフィールドは中間的な状態にならない）。

チェックボックスが3番目の状態となった場合、チェックボックスに関連付けられた変数は値2を返します。

ヒント：入力モードにおいて、「3ステート」チェックボックスは各状態の表示を「チェックなし／チェック／中間状態／チェックなし」のような順序で行います。一般的に、この中間状態は入力モードではあまり役に立たないため、2という値になった場合は、コード上で変数の値を強制的に0に設定し、チェックされた状態からチェックなしの状態へ直接移行してください。

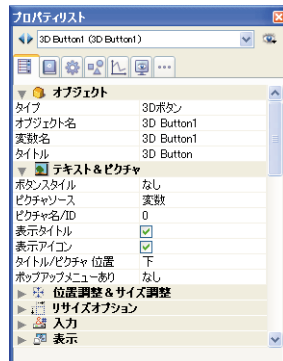
新しい3D ボタン

4th Dimension 2004では、3Dボタンファミリーに対して利用できる機能が増えています。この変更は、次の3つのタイプの3Dボタンに関連しています。

- 3D ボタン
- 3D ラジオボタン
- 3D チェックボックス

これらの新機能により、さまざまな定義済スタイルを3D ボタン（ベベルボタン、プッシュボタンなど）に適用したり、あるいはポップアップメニューを関連付けることができます。

新しい「プロパティリスト」のプロパティを使用すると、数値タイプの3D ボタンを利用できます。



新しい3D ボタンプロパティ

互換性に関する注意：ボタンスタイルの「バックグラウンドオフセット」が選択されていると、大部分の新しいプロパティが利用できません。このスタイルは以前の3D ボタンの動作に対応しており、変換後のデータベースの3D ボタンに対してデフォルトとして選択されています。

タイトル

この新しいプロパティを使用すると、ボタンにラベル（タイトル）を設定することができます。このラベルのフォントとスタイルは、「テキスト」テーマで定義します。“¥”記号を使用して、ラベル内で強制的に改行することができます。

任意の位置で改行できる
ボタンラベル

ラベルに“¥”記号を挿入するには、“¥¥”と入力します。

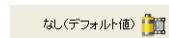
デフォルトとして、ラベルはボタンの中央に配置されます。また、ボタンにアイコンが含まれる場合、「タイトル／ピクチャ位置」プロパティを使い、これら2つの要素の相対位置を変更することができます。

さらに、「タイトル表示」プロパティの選択を解除するとタイトルを隠すことができます。この場合、アイコンは自動的にボタンの中央へ移されます。

ボタンスタイル

このプロパティを使用すると、ボタンの全般的な外観を設定することができます。また、スタイルは、特定オプションの有効性にも関与します。次のスタイルを使用することができます。

■ なし（デフォルト値）



Windows



Mac OS

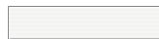
“なし”スタイルが指定された3Dボタンは、透明ボタンと同じです。つまり、ボタンの“強調表示”が行われません。ただし、新しい3Dボタンのオプションは利用できます。

Mac OS上では、リンクしたポップアップメニューを示す三角形を表示できません。

■ バックグラウンドオフセット



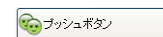
Windows



Mac OS

このスタイルは、以前のバージョンの4th Dimensionで使用された3Dボタンに対応しています。変換後のデータベースの3Dボタンには、このスタイルがデフォルトとして設定されています。「バックグラウンドオフセット」スタイルを選択すると、新しいオプションは一切利用できません。

■ プッシュボタン



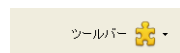
Windows



Mac OS

「プッシュボタン」スタイルが指定された3Dボタンは、標準のシステムボタンとして表示されます。また、このスタイルには「ポップアップメニュー」プロパティ以外の新しい3Dボタンオプションも利用できます。

■ ツールバーボタン



Windows



Mac OS

3D ボタンのこのスタイルは、主としてツールバーの統合を目的としています。

Windows 上では、マウスのロールオーバー時にボタンが強調表示されます。「ポップアップメニュー」プロパティを使用すると、ボタンの右側中央に三角形が表示されます。

Mac OS 上では、ボタンは強調表示されません。「ポップアップメニュー」プロパティを使用すると、ボタンの右下部分に三角形が表示されます。

■ カスタム



Windows

Mac OS

この3D ボタンスタイルは、カスタマイズしたバックグラウンドピクチャに対応し、さまざまな追加パラメータを管理することができます（アイコンやマージンのオフセット）。詳細については、後述の「カスタム3D ボタン」の節を参照してください。

このスタイルには、「ポップアップメニュー」プロパティ以外の新しい3D ボタンオプションを利用することができます。

■ 円



Windows

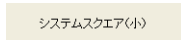
Mac OS

Mac OS X では、「円」スタイルが指定された3D ボタンは円形のシステムボタンとして表示されます。円に対し2種類の規定スタイルが利用できますが、これはフォーム上のボタンのサイズによって決まります。

また、このスタイルには、「ポップアップメニュー」プロパティ以外の新しい3D ボタンオプションを利用することができます。

Windows において、このボタンスタイルは「なし」スタイルと同じです（背景にある円は考慮されない）。

■ スモールシステムスクエア



Windows

Mac OS

Mac OS X では、「スモールシステムスクエア」スタイルが指定された3D ボタンはシステムスクエアとして表示されます。

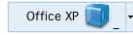
このスタイルには、「ポップアップメニュー」プロパティ以外の新しい3Dボタンオプションを利用することができます。

Windowsにおいて、このボタンスタイルは「なし」スタイルと同じです（背景にあるスクエアは考慮されない）。

■ Office XP



Windows



Mac OS

「Office XP」スタイルが指定された3Dボタンには、次のような特徴があります。

■ 強調表示と背景のカラーはシステムカラーに基づいています。

■ Windows で使用する場合、その強調表示はマウスのロールオーバー時にのみ行われます。

このスタイルには、新しい3Dボタンオプションを利用することができます。

■ ベベル



Windows



Mac OS

Mac OS 上で、「べベル」ボタンは標準のシステムボタンとして表示されます。このボタンには、「ポップアップメニュー」プロパティを含め、新しい3Dボタンオプションを利用することができます。

Windows において、このボタンスタイルは「ツールバーボタン」スタイルと同じですが、ポップアップメニューが関連付けられていることを示す三角形がボタンの右下に表示される点だけが異なります。

■ 角の丸いべベル



Windows




Mac OS

Mac OS において、「角の丸いべベル」ボタンは「べベル」ボタンと同じですが、強調表示されると丸みを帯びた形になる点が異なります。

Windows において、このボタンスタイルは「べベル」スタイルと同じです。

3D ボタンにアイコンを追加する

3D ボタンスタイルにアイコンを追加することができます（「バックグラウンドオフセット」スタイルを除く）。

ツールバー  アイコン

関連付けたアイコンの管理は、複数のプロパティ（ピクチャソース、ピクチャ名/ID、アイコンオフセット）を用いて行います。

■ ピクチャソース

ピクチャボタンと同様に、3D ボタンのアイコンは異なる3種類のソース（変数、ピクチャライブラリ、リソースファイル）から参照します。ソースを設定した後、「ピクチャ名/ID」プロパティにピクチャの名前または番号を指定することができます。

■ ピクチャ名/ID

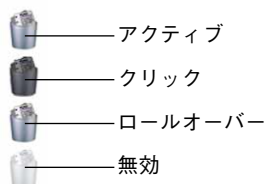
ピクチャソースの設定後、このエリアにピクチャの名前（ピクチャが変数またはピクチャライブラリに存在する場合）、または番号（ピクチャがピクチャライブラリまたはリソースファイルに存在する場合）を入力します。

3D ボタンピクチャを追加したくない場合は、このプロパティに0を渡します。

■ フレーム

ピクチャは、その参照先とは関係なく、4つの個別の縦のエリアを納めなければなりません。これは4th Dimensionがボタンの標準的な4つの状態（アクティブ、ヘルド、ロールオーバー、無効）を表わすために使われます。

例として、4th Dimensionのフォームで使用されるピクチャを次に示します。



注：このエディタにおいて、ピクチャライブラリのピクチャは“フレーム”に納める必要はありません。

■ ピクチャ位置

デフォルトして、ピクチャはボタンの中央に置かれます。ボタンにもラベルがある場合、「タイトル／ピクチャ位置」プロパティを用いてこれら2つの項目の相対的位置を変更することができます。



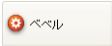


また、「アイコン表示」オプションの選択を解除すると、アイコンを隠すことができます。この場合、テキストは自動的にボタンの中央に配置されます。

タイトル/ピクチャ位置

このプロパティを使用し、関連するアイコンとの関係でボタンのタイトルの相対的位置を変更することができます。ボタンにタイトル（関連ピクチャなし）またはピクチャ（タイトルなし）だけが含まれている場合、このプロパティは無効です。

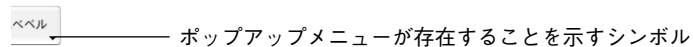
3D ボタンにタイトルとピクチャが含まれる場合、デフォルトとしてテキストはピクチャの下部に置かれます。

このプロパティの各種オプションの結果を次に示します。

- | | |
|---|---|
|  | ■ 左：テキストはアイコンの左側に置かれます。ボタンの内容は右揃えされます。 |
|  | ■ 上：テキストはアイコンの上側に置かれます。ボタンの内容は中央揃えされます。 |
|  | ■ 右：テキストはアイコンの右側に置かれます。ボタンの内容は左揃えされます。 |
|  | ■ 下：テキストはアイコンの下側に置かれます。ボタンの内容は中央揃えされます。 |
|  | ■ 中央：アイコンのテキストはボタンの縦と横に中央揃えされます。例えば、アイコンに納められたテキストの場合に、このパラメータが役立ちます。 |

ポップアップメニュー

この新しいプロパティを使用し、3D ボタン内に三角形として表われるシンボルを表示することができます。このシンボルは、ポップアップメニューが付属することを示します。



このシンボルの外観と位置は、ボタンのスタイルと現在のプラットフォームによって変わります。「ポップアップメニュー」プロパティに対応する3D ボタンのスタイルは次の通りです。

- なし
- ツールバーボタン
- ベベル
- 角の丸いベベル
- Office XP
- リンクありとリンクなし

ポップアップメニューシンボルを3Dボタンに付加する際に、「リンクあり」と「リンクなし」という2つのオプションを使用することができます。



注：実際に「リンクなし」モードを利用できるかどうかは、ボタンのスタイルとプラットフォームによって決まります。

理論上、それぞれのオプションによってボタンと付属のポップアップメニューとの関係が特定されます。

- ポップアップメニューが分かれている場合、ボタンの左側部分をクリックするとボタンのカレントアクションが直接実行されます。このアクションは、ボタンの右側にあるポップアップメニューを用いて変更することができます。
- ポップアップメニューがリンクしている場合、ボタンをクリックしてもポップアップメニューが表示されるだけです。このポップアップメニュー上のアクションを選択しないと実行は行われません。

また、これらのオプションは、ボタンのフォームイベント管理にも影響を与えます（この件に関する詳細は、後述の「3Dボタンにリンクしたフォームイベント」の節を参照）。

■ ポップアップメニューの管理

「ポップアップメニュー」プロパティは、ボタンのグラフィック面を管理するだけであるということに注意してください。

特に、新しいフォームイベント（上記参照）や **Pop up menu** 関数で利用する場合、ポップアップメニューとその値の表示はすべて開発者が処理しなくてはなりません。

カスタム 3D ボタン

「カスタム3Dボタン」を選択すると、さらに複数のプロパティ（バックグラウンドソース、バックグラウンド名/ID、アイコンのオフセット、水平マージン、垂直マージン）が使用できるようになります。

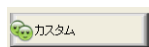
- バックグラウンドソース：このプロパティを使用すると、ボタンの背景に描かれるピクチャを設定することができます。アイコンと同様に、ピクチャの参照先が変数、ピクチャライブラリ、リソースファイルのいずれであるかを指定することができます。
- バックグラウンド名/ID：ソースの設定後、このエリアにピクチャの名前や番号を指定することができます。

アイコンと同様に、バックグラウンドピクチャは4つの個別の縦のエリアを含まなければなりません。これは4th Dimensionがボタンの標準的な4つの状態（アクティブ、ヘルド、ロールオーバー、無効）を表わすために使われます。クリックによりもたらされる結果は、「アイコンのオフセット」プロパティを使用して処理することもできる点に注意してください。

- **アイコンのオフセット**：このプロパティを使用し、独自のオフセット値をピクセル単位で設定することができます。この値はボタンのクリック時に使用され、ボタンのタイトルは入力されたピクセル数だけ右側および下側へシフトされます。

この機能により、ボタンのクリック時に独自の3D効果を適用できます。

- **水平マージン／垂直マージン**：このプロパティを使用し、ボタンの内部マージンのサイズ（ピクセル単位）を設定することができます。これらのマージンにより、テキストならびに3Dボタンアイコンが超えてはならない領域が画定されます。例えば、バックグラウンドピクチャに境界線が含まれる場合に、このパラメータが役立ちます。



マージンなしのカスタム3Dボタン



13ピクセルのマージンが指定された3Dボタン

その他の変更

4th Dimension 2004のフォームオブジェクトに対し、さまざまな変更が行われています。これらの変更については次の節に記載しています。

オブジェクト変数のブールタイプ

ラジオボタンやチェックボックス、標準ボタンに関連付けた変数に対し、ブールタイプを割り当てられるようになりました（**C_BOOLEAN** コマンドを使用）。以前のバージョンの4th Dimensionにおいて、これらの変数は必ず数値タイプでした。

ブールタイプの変数を使用すると、オブジェクトを選択または保留した場合は“True（真）”に、それ以外の場合は“False（偽）”に設定されます。

組み込み参照の計算

以前のバージョンの4th Dimensionにおいて、スタティックテキストエリアに組み込まれた参照（例：“<var_type> [Clients]Name”）は、フォームのロード時に一度計算されていました。

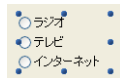
今後、これらの参照は関連付けられた参照を変更すると再計算されるようになります。これは変数の動作と同じです。

新しいラジオボタンの動作

以前のバージョンの4th Dimensionでは、ボタンに関連付けた変数の名前に同じ頭文字を指定することにより、グループ化したラジオボタンの動作を統一することができました(例えば、m_button1、m_button2、m_button3など)。

注：統一した動作とは、グループ化したボタンの中から一度にひとつのボタンしか選択できないということを意味します。

ある状況において、この方法では不十分であることが判明したため、4th Dimension 2004では次のように変更されました。統一した方法で動作するには、「フォーム」エディタ上で一連のラジオボタンがグループ化されていなくてはなりません。グループ化を行うには、「オブジェクト」メニューの「グループ化」コマンド、またはフォーム上の対応するツールバーボタンを使用します。



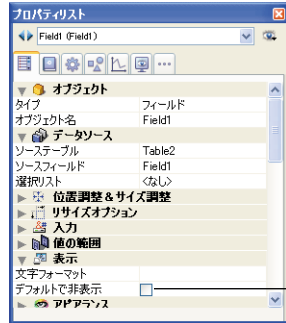
注：セット全体をグループ解除しなくても、グループに属すオブジェクトを選択できるようになりました。これを行うには、そのグループの選択後、オブジェクト上でCtrlキー+クリック (Windows) またはCommandキー+クリック (Mac OS) します。

互換性上の理由から、変換後のデータベースでは従来のモードがデフォルトとして維持されています。「名前でグループ化されたラジオボタン」(アプリケーション「環境設定」の「アプリケーション」テーマの「互換」ページ)の選択を解除すると、強制的に新しいモードを使用することができます。4th Dimension 2004で作成したデータベースでは、このオプションが表示されません。

この新しいモードは、ラジオボタン、3Dラジオボタン、ピクチャラジオボタンに対して有効です。

新しい「デフォルトで非表示」プロパティ

「プロパティリスト」には、「表示」テーマに「デフォルトで非表示」プロパティが用意され、大部分のオブジェクトに対してこのプロパティを使用することができます。



「デフォルトで非表示」プロパティ

このプロパティにより、ダイナミックなインターフェース開発が容易になります。この作業においては、フォームの「On load」イベント中にプログラムからオブジェクトを非表示にした後で、一部のオブジェクトを表示することが頻繁に必要となります。「デフォルトで非表示」プロパティを使用すると、特定オブジェクトをデフォルトとして非表示に設定しておくことにより、このロジックを逆に働かせることができます。その結果、開発者は状況に応じて **SET VISIBLE** コマンドを使用し、これらのオブジェクトを表示するようプログラミングすることができます。

注：このプロパティとポップアップメニューの「非表示」プロパティとを混同しないでください。ポップアップメニューの「非表示」プロパティは、オブジェクトを描画しませんが、オブジェクトはアクティブな状態のままです。

レコードリストの管理

4th Dimension 2004では、レコードリストの管理が変更されてさらに管理しやすくなり、各種タイプのリストの機能も向上しています。

以前のバージョンの4th Dimensionでは、3つの環境においてレコードを表示することができ、各環境ごとに固有のプロパティがありました。

■「ユーザ」モードのリストフォーム

■ **MODIFY SELECTION** コマンドまたは **DISPLAY SELECTION** コマンドを使用して表示されたリストフォーム

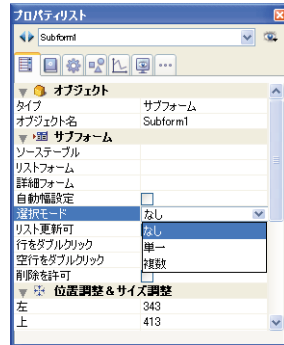
■ サブフォーム（組み込みフォームとも呼ばれる）

4th Dimension 2004では、これら3つの環境の機能が同期化されています。それぞれの機能は実質的には同じですが、以前のバージョンの4th Dimensionとの互換を図る上で必要となる相違点がいくつか存在します。

選択モード

レコードリストは、「なし」、「1レコード」、「複数レコード」という3つの選択モードに対応します。

- サブフォームに関しては、サブフォームオブジェクトを選択すると、「選択モード」プロパティを適用することができます。



- **MODIFY SELECTION** コマンドおよび **DISPLAY SELECTION** コマンドを使用したリストフォームに関しては、新しいパラメータを用いて選択モードを設定することができます（後述の「**MODIFY SELECTION**、**DISPLAY SELECTION**」の節を参照）。

- 「ユーザ」モードのリストフォームでは、自動的に「複数レコード」モードが使われます。

なし

このモードを選ぶと、レコードの選択を行えません。システムセットである UserSet は定義されません。「リスト更新」オプションがアクティブではない場合、リスト上のクリックは無効となります。

移動キーを使用してもリストがスクロールされるだけであり、新しい「On selection change」フォームイベントは生成されません（このフォームイベントに関する詳細は、後述の「レコードリストにリンクしたフォームイベント」の節を参照）。

注：以前のバージョンの 4th Dimension では、「表示のみ」オプションを用いて、サブフォームにこのモードを利用することができました。

1 レコード

このモードでは、一度にひとつのレコードを選択することができます。システムセットである UserSet は定義されません。クリックするとレコードが選択され、それがカレントレコードとなります。Ctrl キー+クリック (Windows) または Command キー+クリック (Mac OS) のキーコンビネーションをレコード上で使用すると、その状態が切り替わりません (選択または非選択)。

「次」および「前」矢印キーにより、リスト上の前/次レコードを選択します。他の移動用キーはリストのスクロールを行います。カレントレコードが変更されるたびに、「On selection change」フォームイベントが新たに生成されます。

注：以前のバージョンの 4th Dimension では、「選択可」オプションを用いてサブフォームでこのモードを利用することができました。

複数レコード

このモードでは、複数のレコードを同時に選択することができます。システムセットである UserSet は使用可能です。クリックするとレコードが選択されますが、これによりカレントレコードが変更されることはありません。Ctrl キー+クリック (Windows) または Command キー+クリック (Mac OS) のキーコンビネーションをレコード上で使用すると、その状態が切り替わりません (選択または非選択)。Ctrl キー+クリック (Windows) または Command キー+クリック (Mac OS)、ならびに Shift キー+クリックのキーコンビネーションを使用すると、複数レコードを選択することができます。

「次」および「前」矢印キーにより、リスト上の前/次レコードを選択します。他の移動用キーはリストのスクロールを行います。選択レコードが変わるたびに、「On selection change」フォームイベントが新たに生成されます。

注：以前のバージョンの 4th Dimension では、MODIFY SELECTION コマンドと DISPLAY SELECTION コマンドを使用して表示されたリストに対してのみ、このモードを利用することができました。

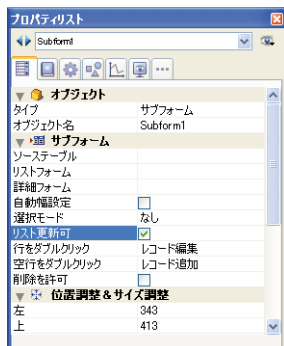
HIGHLIGHT RECORDS コマンドや新しい **GET HIGHLIGHTED** コマンドを使用すると、プログラムから選択レコードを管理することができます。

フォームメソッドやオブジェクトメソッドが実行されると、4th Dimension はカレントセレクションが変更されたかどうかを検証します。変更が行われ、かつ **HIGHLIGHT RECORDS** コマンドが呼び出されなかった場合、4th Dimension はレコードセレクションを初期化して「On selection change」フォームイベントを生成します。

リスト更新

あらゆるタイプのレコードリストに対し、「リスト更新」モードを利用できるようになりました。今までは、「ユーザ」モードでのみこのモードを使用できました。

- サブフォームに関しては、サブフォームオブジェクトを選択すると、「リスト更新」モードを適用することができます。



- **MODIFY SELECTION** コマンドと **DISPLAY SELECTION** コマンドを使用したリストフォームについては、新しいパラメータを用いて選択モードを設定することができます（後述の「**MODIFY SELECTION、DISPLAY SELECTION**」の節を参照）。

- 「ユーザ」モードのリストフォームでは、「リスト更新」プロパティが自動的に有効になります。

「リスト更新」モードでは、入力フォームを使わなくてもリスト上のレコードデータを直接修正することができます。修正を行うには、編集するフィールドをクリックします。すべてのリストに対し、デフォルトとして「リスト更新」モードが有効になっています。

ユーザが行をクリックして編集可能な状態にすると、対応するデータがロードされます。フォームとオブジェクトメソッドが呼び出され、新しい「On selection change」フォームイベントが生成されます（後述の「レコードリストにリンクしたフォームイベント」の節を参照）。

リストの表示

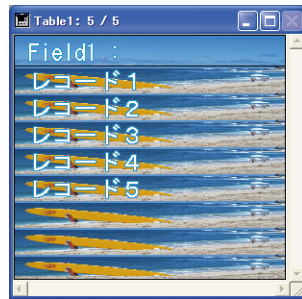
4th Dimension 2004では、レコードリスト表示のメカニズムが最適化されています。次の事柄が変更されています。

- リストフォームのボディエリアにあるオブジェクトの表示
- 空の行の表示
- 選択状況に応じたオブジェクトの表示

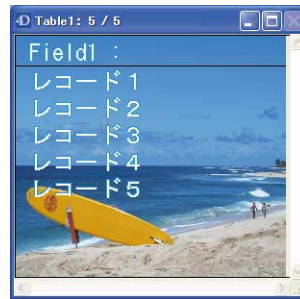
- フォームイベントの起動
- リストフォームの各エリアで受け入れられるアクティブオブジェクト

フォームのボディにあるオブジェクト

上部がヘッダエリアから開始し、下部がブレイクエリアで終わるオブジェクトは、オブジェクトの高さを大きくすることにより一度だけ描画されるようになります。今まで、ボディエリアに位置する部分は反復して表示されていました。次の図に示すように、ピクチャに関してのみこの変更が明らかになります。



4th Dimension 2003



4th Dimension 2004

このメカニズムでは、ピクチャ表示に関するプロパティが考慮されます（切り捨て、モザイクなど）。

空の行

リストモードでは、必要に応じて4th Dimensionは、ウィンドウ全体を埋めるように最後の表示レコードの後ろに空の行を表示します。

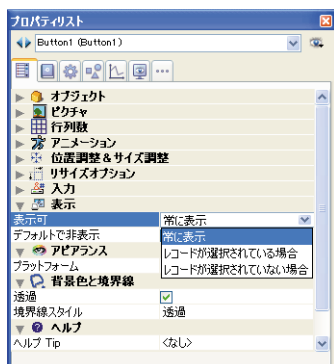


空の行

また、このウィンドウが表示されると、リストの空の各行に対して「On display detail」フォームイベントが生成されます。この状況において、カレントレコードは存在しません。**Record number**関数は-1を返し、**Selection number**関数は有効な値を返しません。この場合、新しい**Displayed line number**関数を使用すると、現在表示されている行数を知ることができます。

オブジェクトの表示

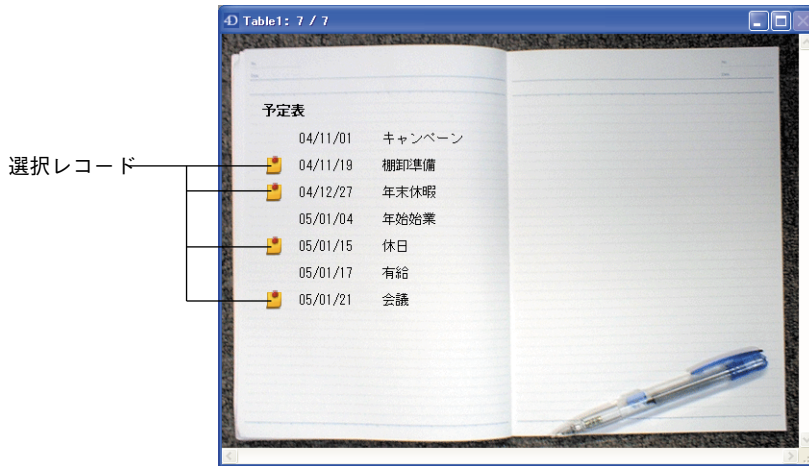
フォームがリストフォームである場合、「フォーム」エディタの「プロパティリスト」において、新しい表示用プロパティである「表示」をすべてのオブジェクトに対して使用することができます。



このプロパティは3つの値を受け入れます。

- 常に表示
- レコード選択時
- レコード非選択時

このプロパティは、リストフォームのボディ部分に配置されたオブジェクトを描画する際にのみ使用されます。このプロパティは、処理中のレコードの選択状況に応じ、オブジェクトの描画を行うか、または行わないかを4th Dimensionに通知します。このプロパティを使用すると、カラー以外の表示属性を用いて、選択レコードを表わすことができます。



SET VISIBLE コマンドを使用してオブジェクトが隠されている場合、4th Dimensionはこのプロパティを考慮しません。この場合、レコードセレクションが選択されているかどうかに関わらず、オブジェクトは非表示のままです。

最終レコードの後ろに空の行を表示する場合（前述の「空の行」の節を参照）、4th Dimensionは「常に表示」プロパティまたは「レコード非選択時」プロパティを用いて、空の各行にオブジェクトを描画します。

フォームイベントの呼び出し順序

4th Dimension 2004において、サブフォームのメソッドとフォームイベントの呼び出し順序は、**MODIFY SELECTION** コマンドや **DISPLAY SELECTION** コマンドにより表示されたリストフォームの呼び出し順序に合わせることができました。次の節では、以前のバージョンとの相違点について説明します。

■ 4th Dimension 2004以前のサブフォーム呼び出し

各レコードに対して：

ファイルプロシージャ（古いスキーム）

「On display detail」 イベントが指定されたフォームメソッド

■ 4th Dimension 2004のサブフォーム呼び出し（= **MODIFY SELECTION** または **DISPLAY SELECTION** で表示されたリストフォームの呼び出し）

ヘッダエリアの各オブジェクトに対して：

「On header」 イベントが指定されたオブジェクトメソッド

「On header」 イベントが指定されたフォームメソッド

各レコードに対して：

ボディ内の各オブジェクトに対して：

「On display detail」 イベントが指定されたオブジェクトメソッド
「On display detail」 イベントが指定されたフォームメソッド

サブフォームに対して、v3.x.xのファイルプロシージャが呼び出されなくなったことに注意してください。

アクティブオブジェクト

リストフォームのヘッダ、ブレイク、フッタ上のアクティブオブジェクトの管理が拡張されています。

■ **MODIFY SELECTION** または **DISPLAY SELECTION** により表示されたリストフォームおよびサブフォームは、ヘッダやブレイク上のあらゆるアクティブオブジェクトを受け入れます。

■ **MODIFY SELECTION** または **DISPLAY SELECTION** により表示されたリストフォームは、フッタ上のあらゆるアクティブオブジェクトを受け入れます。

サブフォームオプション

サブフォームに使用できる属性が追加され、外観や機能をより制御しやすくなります。

水平スクロールバー

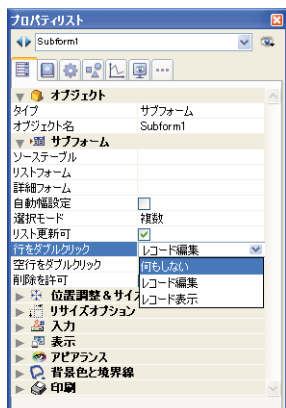
「プロパティリスト」の新しいオプション（「アピランス」テーマ）を使用すると、サブフォームで水平スクロールバーを利用できるようになります。このオプションを選択すると、本来のサブフォームエリアの外側に水平スクロールバーが追加されます。

ダブルクリックアクション

ユーザのダブルクリックに対するサブフォームの動作を決定するパラメータを設定できるようになりました。また、以前のバージョンの4th Dimensionで作成されたデータベースの場合、空の行で行われたダブルクリックへの対応を設定することができます。

これらのパラメータは、「プロパティリスト」の「サブフォーム」テーマで利用することができます。

ダブルクリック
管理オプション



■ 行でダブルクリック：サブフォームレコード上でダブルクリックが行われた場合に実行する動作。次のようなオプションを利用できます。

■ 何もしない：ダブルクリックを無視します。

■ レコード編集：サブフォームレコードを編集モードに移行します。「リスト更新」オプションが選択されている場合、直接リスト上で変更が行われます。「リスト更新」オプションが選択されていない場合は、ページモードで変更が行われ、サブフォームに関連付けられた詳細フォームが使用されます。

■ レコード表示：サブフォームに関連付けた詳細フォームを用いて、ページモードでレコードデータを表示します（リードオンリー）。

■ 空の行でダブルクリック（変換後のフォームのみ）：以前のバージョンの4th Dimensionでは、サブフォームの空白行でダブルクリックを行うことにより、サブフォームへレコードを追加していました。互換性上の理由から、このメカニズムは以前のバージョンの4th Dimensionで作成されたフォームに対してのみ維持されています。

しかしながら、標準のアクションを使用してレコードを追加することをお勧めします（後述の「標準アクション」の節を参照）。

サブフォームの空白行でダブルクリックが行われた場合に実行する動作を定義することができます。次のようなオプションを利用できます。

■ 何もしない：ダブルクリックを無視します。

■ レコード追加：サブフォーム上で新規レコードを作成し、編集モードへ移行します。「リスト更新」オプションが選択されている場合、直接リスト上にレコードが作成されます。「リスト更新」オプションが選択されていない場合は、ページモードでレコードが作成され、サブフォームに関連付けられた詳細フォームが使用されます。

サイズ調整

サブフォームのサイズ調整に関するメカニズムが変更され、「プロパティリスト」で設定したオプションをより操作しやすくなりました。

フォーカス

サブフォームオブジェクトに対して、「フォーカス表示」プロパティを適用できるようになりました。サブフォームがフォーカスを取得すると、ユーザは移動用キーや「すべてを選択」メニュー（複数行を選択している場合）を用いて、それをコントロールすることができます。

サブフォームがフォーカスを取得するか、またはフォーカスを失った場合、「On getting focus」イベントや「On losing focus」イベントを用いて、親フォームのフォームメソッドが呼び出されます。この際、**Focus object**関数（**Last object**関数の新しい名前）は、サブフォームのテーブルまたはサブテーブルへのポインタを返します。

サブフォームでは、フォーカスを取得したことを示す点滅三角形が表示されなくなります。今後この情報は、選択されたサブレコードの色を変えることにより表わされます。また、独自の画像インジケータを利用することもできます（例えば、「On getting focus」イベントを使用）。

標準アクション

■ 4th Dimension 2004では、標準アクションである「サブレコード追加」と「サブレコード編集」の動作が変更されました。

サブフォーム以外に、これらのアクションは**MODIFY SELECTION**または**DISPLAY SELECTION**コマンドならびに「ユーザ」モードで表示されたリストフォームでも機能します。

レコードの追加や変更は、リスト上で、またはページ上で、フォームの「リスト更新」オプションのステータスに応じて実行されます。このオプションは、**MODIFY SELECTION**または**DISPLAY SELECTION**コマンドの<入力リスト>引数を使用してサブフォームプロパティに、または「レコード」メニュー（「ユーザ」モード）で設定されます。

レコードの追加や変更は、そのフォームの「リスト更新」オプションのステータスに応じて、リスト上またはページ上のいずれかで実行されます。このオプションは、**MODIFY SELECTION**または**DISPLAY SELECTION**コマンドの<入力リスト>引数を使用してサブフォームプロパティに設定するか、または「レコード」メニュー（「ユーザ」モード）を使用して設定します。

- リスト入力可能な場合、直接リスト上でレコードの追加や変更が実行されます。
 - リスト入力が行えない場合、ページモードでレコードの追加や変更が実行されます。
- これらのアクションは、ボタンやメニューコマンドに対して使用することができます。

オブジェクトライブラリ

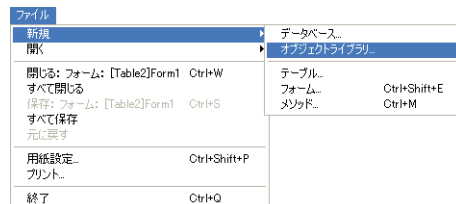
4th Dimension 2004では、オブジェクトライブラリの作成や利用ができます。

オブジェクトライブラリとは、4th Dimensionにより作成される外部ファイルです。これを使用して、ボタン、テキスト、ピクチャ、階層リストなど、フォーム上で使用するあらゆるタイプのオブジェクトを保存することができます。各オブジェクトは、オブジェクトメソッドを含めて、そのプロパティとともに保存されます。各ライブラリはひとまとめにされ、ドラッグ&ドロップまたはコピー&ペーストだけで使用することができます。これは常設のクリップボードのようなものです。

ライブラリを使用すると、グラフィックファミリーや動作などによりグループ化されたフォームオブジェクトの背景を作成することができます。これらのライブラリは外部ファイルとして保存されるため、さまざまなデータベースから簡単に利用することができます。

オブジェクトライブラリを作成する

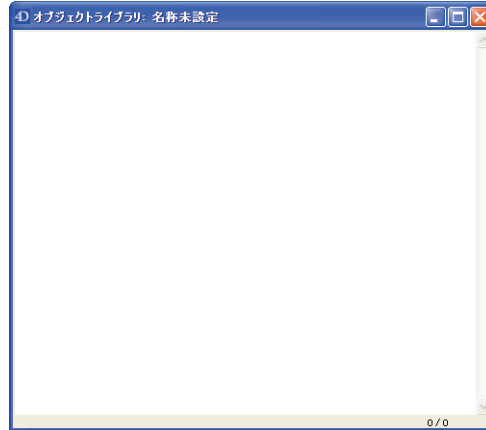
新しいオブジェクトライブラリを作成するには、4th Dimensionの「ファイル」メニューから「新規>オブジェクトライブラリ...」コマンドを選択します。



標準の「ファイル保存」ダイアログボックスが表示され、オブジェクトライブラリの名前と保管場所を選択することができます。Windowsにおいて、ライブラリの拡張子は“.4il”です。

注：この拡張子は、4D Insiderのオブジェクトライブラリと同じです。実際に、これら2つのファイルタイプは類似しています。

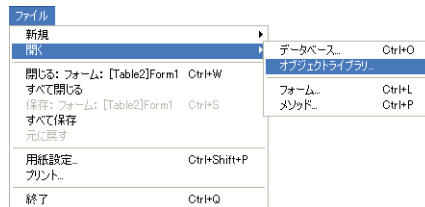
ダイアログボックスを確定すると、4th Dimensionは新しいオブジェクトライブラリをディスク上に作成し、そのウインドウを表示します（デフォルトでは空）。



データベースごとにライブラリをいくつでも作成することができます。Mac OS上で作成され構築されたライブラリは、Windows上でも使用可能です（その逆も同様）。

オブジェクトライブラリを開く

既存のオブジェクトライブラリを開くには、4th Dimensionの「ファイル」メニューから「開く>オブジェクトライブラリ...」コマンドを選択します。



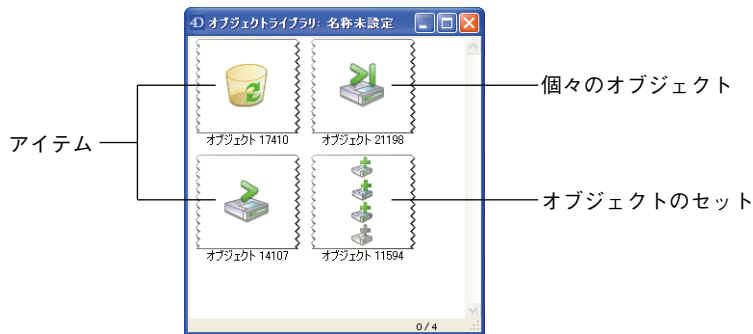
標準の「ファイルを開く」ダイアログボックスが表示され、開こうとするオブジェクトライブラリを選択することができます。

特定のオブジェクトライブラリは、一度にひとつのデータベースでしか開くことができません。ただし、ひとつのデータベースからは複数の異なるライブラリを開くことができます。

オブジェクトライブラリの構築

ドラッグ&ドロップまたはカット&コピー&ペーストを使用し、オブジェクトをオブジェクトライブラリに配置します。オブジェクトは、フォームまたは他のオブジェクトライブラリいずれのものでも構いません。元のオブジェクトとのリンクは維持されないため、元のオブジェクトが変更された場合でも、コピーしたオブジェクトへの変更は行われません。

個々のオブジェクト、またはオブジェクトのセットをオブジェクトライブラリに納めることができます。オブジェクトやセットはそれぞれ単独のアイテムにまとめられます。各アイテムはフォームとして見なされます。



オブジェクトライブラリには、32,000個までのアイテムを納めることができます。

オブジェクトは、グラフィックと機能の双方に関するすべてのプロパティとともにコピーされます。アイテムがフォームや別のライブラリにコピーされる際も、これらのプロパティが完全に保持されます。

各新規アイテムには、「Object2012」のように“Object”の後ろに乱数を付加した名前が付けられます。この項目名は自由に変更することができます。項目名を変更するには、次の手順を実行します。

- 名前を変更するアイテムをダブルクリックする。
- 名前を変更するアイテムを右クリック（Windows）またはControl+クリックし、コンテキストメニューより「名前の変更」コマンドを選択します。



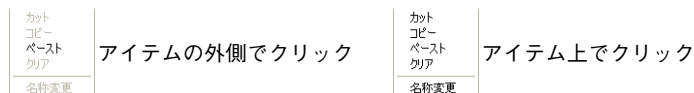
ダイアログボックスが表示され、アイテム名を変更することができます。



複数のアイテムに同じ名前を付けることができます。

コンテキストメニュー

各オブジェクトライブラリにはコンテキストメニューがあり、このメニューを用いて基本的な管理コマンドにアクセスすることができます。このメニューの内容は、マウスクリックを行う場所によって変わります。



注：複数アイテムを同時に選択することはできません。

- **カット／コピー**：これらのコマンドを使用し、オブジェクトライブラリの選択アイテムをメモリ上に格納することができます。「カット」コマンドは、ライブラリからそのアイテムを削除します。いずれの場合も、この後でアイテムをフォームや別のライブラリにペーストすることができます。
- **ペースト**：このコマンドは、ライブラリと互換性があるオブジェクトがメモリ上に存在する場合にのみアクティブになります。このコマンドを使用して、メモリ上のオブジェクトをオブジェクトライブラリに挿入することができます。

注：「編集」メニューならびに標準のキーボードショートカットを使用して、これらのコマンドを利用することもできます。

- **消去**：このコマンドは、選択したアイテムをライブラリから削除します。「Del」または「Backspace」キーを使用することもできます。
- **名前の変更**：このコマンドは、アイテム名を変更するダイアログボックスを表示します（前述）。

従属オブジェクト

特定のライブラリオブジェクトに対してコピー&ペーストやドラッグ&ドロップを実行すると、従属するオブジェクトも一緒にコピーされます。例えば、ボタンをコピーすると、付随するオブジェクトメソッドがあればそれも一緒にコピーされます（必須）。これらのオブジェクトは、直接またはドラッグ&ドロップを用いてコピーすることはできません。

次のリストは、それを使用するメインオブジェクトと一緒にライブラリにペーストされる従属オブジェクトを示します（適用可能な場合）。

- リスト
- スタイルシート
- フォーマット/フィルタ
- ピクチャ
- ヒント（フィールドにリンク）
- STR#リソース
- オブジェクトメソッド

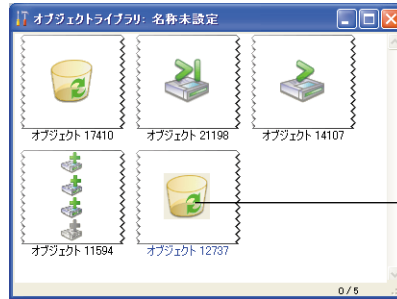
STR#リソースに関しては、そのオブジェクトに関連付けられた文字列だけではなく、STR#リソース全体がコピーされます。

オブジェクトライブラリの表示

オブジェクトライブラリには4th Dimensionの「デザイン」モードからアクセスしますが、開発者が許可する場合は、「ユーザフォーム」エディタからもアクセスすることができます（前述の「オブジェクトライブラリへのアクセス」の節を参照）。

「デザイン」モードでは、ライブラリのすべてのアイテムが表示されます。しかし、「ユーザフォーム」エディタでは、オブジェクトメソッドが関連付けられていないアイテムだけが表示されます。

少なくとも1つのオブジェクトメソッドが関連付けられたアイテムは、バッジ付きで表示され（フォームでは、オブジェクトメソッドを持つオブジェクト上に同じものが表示されます。前述の「バッジを使用する」の節を参照）、その名前が青色で表示されます。



オブジェクトメソッドが付属するアイテム

これにより、「デザイン」モードにおける開発者用の表示アイテムと「ユーザ」モードにおけるユーザ用の表示アイテムを区別することができます。

クライアント/サーバの対応

すべてのクライアントマシンからオブジェクトライブラリへアクセスできるようにするには、「Extras」という名前のフォルダにライブラリファイルを格納し、4D Serverにより開かれるデータベースストラクチャと同じ階層にこのファイルを配置しなくてはなりません。

4D Clientへの接続時に、「Extras」フォルダはクライアントマシンへ送信され、システムのアクティブな4Dフォルダ内にある「Plugins」フォルダと同じ階層に配置されます。

サーバおよびクライアントのプラットフォームに関係なく、この対応が行われます。

アプリケーションビルダ

4th Dimension 2004にはさらに強力な新しいアプリケーションビルダが導入され、クライアント/サーバアプリケーションの配備をはじめ、さまざまなニーズに対応します。

新しいアプリケーションビルダを使用して、次の事柄を実行できます。

- 一つのコンパイル済データベースを元にして、XMLプロジェクトを用いた各種アプリケーションを構築する。
- カスタマイズしたクライアント/サーバアプリケーションを構築する。
- クライアント側の自動更新を伴うクライアント/サーバアプリケーションを構築する。

配布用アプリケーションの新しい名称

シングルユーザ版アプリケーション配布の可能性を明確にし、名称の統一を図るため、4th Dimension 各種製品のうち次のアプリケーションの名称が変更されました（それぞれの用途は以前と同じ）。

- 4D Runtime は 4D Runtime Interpreted に変わります。

インタプリタモードで 4D データベースを実行するには、4D Runtime Interpreted を使用します（このアプリケーションは無償で提供され、無制限に再配布することが可能）。

- 4D Runtime Classic は 4D Runtime Single User に変わります。

コンパイル済の 4D データベースを実行するには、4D Runtime Single User を使用します。

- 4D Engine は 4D Runtime Volume License に変わります。

スタンドアロン版の実行形式アプリケーションを作成するために、コンパイル済の 4D データベースへ 4D Runtime Volume License を組み込むことができます。4th Dimension 2003 と同様に、4D Runtime Volume License には次の種類があります：4D Runtime Volume License Pro、4D Runtime Volume License Sponsored、4D Runtime Volume License Light

これらのアプリケーションの購入や使用に関する条件については、4D 社までお問い合わせいただくか、または弊社の Web サイト（www.4d-japan.com）を参照してください。

ダイアログボックスとアプリケーションプロジェクトの構築

新しい「アプリケーションビルド」ダイアログボックスには、複数のタブが用意されています。



- 「スタンドアロン」タブを使用し、スタンドアロン版のアプリケーション設定を行うことができます（以前のバージョンのアプリケーションビルダ¹と同じ）。
- 「クライアント／サーバ」タブを使用すると、クライアント／サーバ版のアプリケーション設定を行うことができます。
- 「プラグイン」タブを使用すると、アプリケーションに統合するプラグインを設定することができます。
- 「ライセンス」タブを使用すると、使用するシリアル番号を定義することができます。

アプリケーションの各種“ターゲット”パラメータ（スタンドアロンおよびクライアント／サーバ）を同時に設定することができます。チェックボックスを使用して、それぞれを明示的に選択しなくてはなりません。「ビルド」ボタンをクリックすると、選択したすべてのターゲットに対応するアプリケーションが構築されます。

このウィンドウの各パラメータは、XMLキーとして「`buildapp.XML`」という名前のアプリケーションプロジェクトファイルに保存され、データベースの初期設定フォルダ内の「`BuildApp`」サブフォルダに配置されます。初めてこのダイアログボックスを使用する際は、デフォルトのパラメータが使用されます。「ビルド」ボタンをクリックすると、必要に応じてプロジェクトファイルの内容が更新されます。

クライアント側では、XMLファイルはクライアントアプリケーションの「`4D Extensions`」フォルダ内に保存されます。

同じデータベースに対し、別のXMLプロジェクトを複数定義することができます。また、これらのプロジェクトは新しい**BUILD APPLICATION**コマンド（後述）で利用されます。

XMLキーにより、「アプリケーションビルド」ダイアログボックスに表示されるオプションに加えて、さらに次のようなオプションが提供されます。

- サーバのIPアドレスやTCPポート番号の設定
- クライアントアプリケーションとサーバアプリケーション間で互換するバージョン番号の範囲の設定（例えば、クライアントアプリケーションのバージョン1.1から1.3は、サーバアプリケーションのバージョン1.3へ接続可能）

これらのキーに関する説明は、『4D XMLキー』という別のマニュアルに記載されていません。

1. 明確にするため、“ライセンス版”と“デモ版”の実行形式アプリケーション作成オプションは、それぞれ“4D Runtime Volume Proまたは4D Runtime Volume Sponsoredを使用”と“4D Runtime Volume Lightを使用”という名称に変わりました。

ログファイル

はじめてアプリケーションを構築する際に、4th DimensionはログファイルをXML形式で生成します。各アプリケーションプロジェクトごとに、ひとつのログファイルが生成されます。ログファイルには「プロジェクト名.log」という名前が付けられ（“プロジェクト名”はアプリケーションプロジェクトの名前）、アプリケーションプロジェクトファイルと同じ階層に配置されます。

構築を行うたびに、ログファイルには次の情報が保存されます。

- プロジェクトファイルの名前とフルアクセスパス
- データベースの名前とフルアクセスパス
- 構築した日付と時間
- 構築が正常に終了したかどうかを示す注記
- 各構築時間（例えば、スタンドアロンアプリケーション）および処理の合計時間
- 正常終了した構築と不成功の構築の数

クライアント/サーバアプリケーションを構築する

4th Dimension 2004では、同種、クロスプラットフォーム対応、自動更新オプションを備えた独自のクライアント/サーバアプリケーションを構築することができます。

クライアント/サーバアプリケーションとは？

クライアント/サーバアプリケーションは、3つのアイテムの組み合わせにより成り立ちます。

- コンパイル済4Dデータベース
- 4D Server アプリケーション
- 4D Client アプリケーション（Mac OSやWindows）

構築後、クライアント/サーバアプリケーションは、サーバパート（ユニーク）とクライアントパート（各クライアントマシンヘインストール）という独自の2つの部分で構成されます。比較するために、4D Serverを使用して標準的な配備を行う場合は、4D Serverアプリケーション、データベースストラクチャファイル、データベースのデータファイルならびに4D Clientアプリケーションが必要となることを覚えておいてください。

また、クライアント/サーバアプリケーションはカスタマイズされ、その操作が簡略化されています。

■ サーバパートを起動するには、サーバアプリケーションをダブルクリックします。ストラクチャファイルを選択する必要はありません。

■ クライアントパートを起動するには、クライアントアプリケーションをダブルクリックします。すると、直接サーバアプリケーションへ接続します。接続ダイアログボックスでデータベースを選択する必要はありません。

接続が不成功に終わった場合、標準の接続用ダイアログボックスにサーバアドレスを入力することができます。

クライアントパートだけが、対応するサーバパートへの接続を行えます。ユーザが標準の4D Clientアプリケーションを使用して、サーバパートへ接続しようとすると、接続できないことを示すエラーメッセージが返されます。

最後に、ネットワーク経由でクライアントパートが自動更新されるようにクライアント／サーバアプリケーションを設定することができます。この機能については、後述する「クライアントアプリケーションの自動更新」の節で説明します。

アプリケーションパラメータを設定する

“クライアント／サーバアプリケーション”のターゲットは、「アプリケーションビルド」ダイアログボックスの「クライアント／サーバ」ページで設定することができます。



■ “クライアント／サーバアプリケーション”を構築する：このオプションを選択し、アプリケーションビルドにクライアント／サーバ版を含めるかどうかを指示します。これを実施するには、4D Serverアプリケーションの場所と、少なくとも一つの4D Clientアプリケーションの場所を次に指定しなくてはなりません。

■ クライアントアプリケーションの自動更新を許可する：このオプションは、4D Server と 4D Client のアクセスパスが入力された場合にのみアクティブになります。これは、構築後のアプリケーションが、ネットワーク経由でクライアントパートの更新メカニズムを処理することを示します。このメカニズムについては、前述の「クライアントアプリケーションの自動更新」の節で説明しています。

■ 4D Server フォルダ：このエリア（必須項目）は、アプリケーション構築に使用する 4D Server アプリケーションを含むフォルダへのアクセスパスを指示します。そのバージョンは、アプリケーションビルダが実行されるプラットフォームに適合していません。

4D Server フォルダを選択するには、「...」 ボタンをクリックしてフォルダ検索ダイアログボックスを使用し、4D Server アプリケーションを探します。

■ 4D Client フォルダ (Windows)：このエリア（任意項目）は、アプリケーション構築に使用する 4D Client アプリケーションの Windows 版を含むフォルダへのアクセスパスを指示します。クライアントパートの Windows 版を作成したくない場合は、このオプションを選択しないでください。

Windows 版 4D Client のフォルダを選択するには、「...」 ボタンをクリックして標準のダイアログボックスを使用し、4D Client アプリケーションを探します。

■ 4D Client フォルダ (Mac OS)：このエリア（任意項目）は、アプリケーション構築に使用する 4D Client アプリケーションの Mac OS 版を含むフォルダへのアクセスパスを指示します。クライアントパートの Mac OS 版を作成したくない場合は、このオプションを選択しないでください。

Mac OS 版 4D Client のフォルダを選択するには、「...」 ボタンをクリックして標準のダイアログボックスを使用し、4D Client アプリケーションを探します。

4D Server および 4D Client フォルダをカスタマイズする

実行形式のクライアント/サーバアプリケーションを構築する際、4th Dimension は 4D Server フォルダの内容を作成先フォルダの「Server」サブフォルダに複製し、4D Client フォルダの内容を作成先フォルダの「Client」サブフォルダに複製します。この後、必要に応じてオリジナルの 4D Server と 4D Client フォルダの内容をすべてカスタマイズすることができます。

例えば、次のような事柄を実行できます。

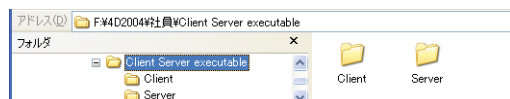
- 特定の言語に対応する 4D Server バージョンをインストールする。
- 「PlugIns」 フォルダへファイルやフォルダを追加する。
- 「4D Extensions」 フォルダの内容をカスタマイズする。

注：Mac OSにおいて、4D Serverはパッケージとして提供されます。まず始めにパッケージの内容を表示してから（アイコンをControl+クリック）、変更を行うことができます。

生成されるファイル

クライアント／サーバアプリケーションが構築されると、作成先フォルダ内に「Client Server Engine」という名前のフォルダが新しく追加されます。

このフォルダには、「Server」と「Client」という2つのフォルダが含まれています。



注：エラーが発生した場合、これらのフォルダは生成されません。この場合、ログファイルをオープンして（前述の「ログファイル」の節を参照）エラーの原因を調べてください。

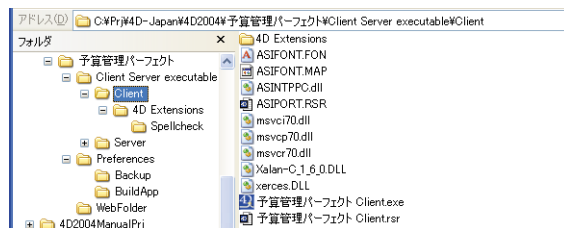
「Client」フォルダには、アプリケーションビルダの指定したプラットフォームに対応するアプリケーションのクライアントパートが含まれます。各フォルダには、各クライアントマシンへインストールしなくてはなりません。

「Server」フォルダには、サーバアプリケーションおよびアプリケーションが機能するために必要な項目が納められます。

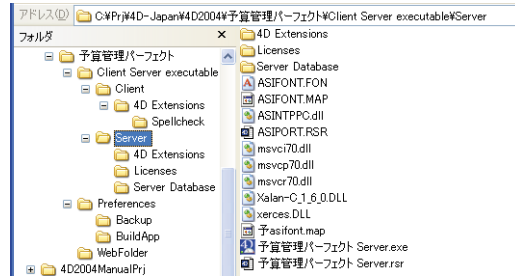
各フォルダの内容は現在のプラットフォームに大きく依存します。

Windows：このフォルダに含まれるのは、“アプリケーション名.exe”という名前のクライアントアプリケーション実行形式ファイル（および対応する.rsrファイル）、アプリケーションが機能するために必要となる各種ファイルとフォルダ、ならびにオリジナルの4D Clientフォルダに含まれる独自の項目です。

クライアントフォルダ



サーバフォルダ



Mac OS：このフォルダには、“アプリケーション名”という名前のクライアントアプリケーションパッケージが納められます。その中には、アプリケーションが機能するために必要な項目がすべて含まれます。Mac OSでは、このパッケージをダブルクリックして起動します。



注：生成された Mac OS パッケージには、Windows のサブフォルダと同じ項目が納められます。パッケージの内容を表示して（アイコン上で Control+ クリック）、項目を修正することができます。

また、「クライアントアプリケーションの自動更新を許可する」オプションを選択した場合、「Server」フォルダには Mac OS や Windows 形式のクライアントアプリケーション（「Client」フォルダの内容）が圧縮ファイルとして納められます。クライアントアプリケーションを自動更新する際に、このファイルが使用されます（前述の「クライアントアプリケーションの自動更新」の節を参照）。

注：以前のバージョンと同様に、新たに構築を行う前に目的のフォルダの全内容が削除されます。アプリケーションの一部分だけを生成する場合でも（例えば、クライアントパート）、このメカニズムが適用されます。したがって、新しいバージョンを作成する前に、目的のフォルダの中で残したい項目は必ず他へ移すようにしてください。

プラグイン管理

新しいアプリケーションビルダを使用し、アプリケーションで使用するプラグインを指定することができます。これを行うには、「プラグイン」タブをクリックし、利用可能なプラグインのリストを表示します。



このリストには現在の4th Dimension アプリケーションによりロードされるプラグインが表示されます。「アクティブ」カラムは、そのプラグインが構築後のアプリケーションに組み込まれることを示します。デフォルトでは、すべてのプラグインが選択されています。プラグインの選択を解除するには、関連するオプションをクリックしてください。

同じプラグインが4th Dimension によりロードされ、4D Engine や4D Server にもインストールされている場合に、そのプラグインが同時に選択されると、4D Engine や4D Server へインストールされたプラグインが優先されます。つまり、この場合は4D Engine や4D Server へインストールされたプラグインが組み込まれます。ソースアプリケーションフォルダの内容をコピーする手法は実行可能なアプリケーションに様々なタイプのファイルを統合するために使うことができます。

注：4th Dimension 2004 におけるプラグインのインストールに関する詳細は、前述の「新しいプラグインアーキテクチャ」の節を参照してください。

2つの異なるバージョンの同じプラグインでコンフリクトが生じる場合（1つは4th Dimension が読み込み、もう一方はソースアプリケーションフォルダに存在）、「4D Runtime Volume License」または「4D Server」フォルダにインストールされたプラグインが優先されます。

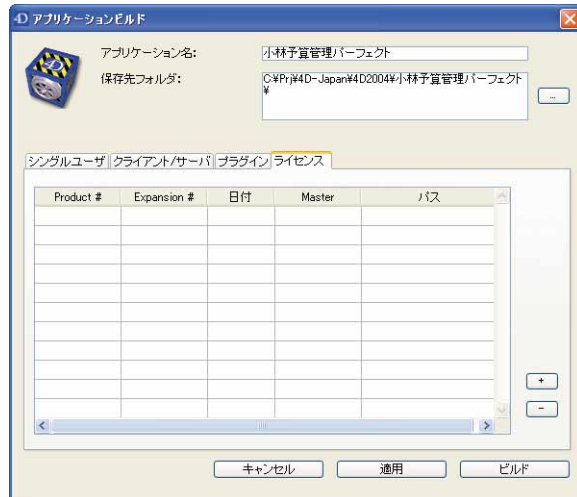
注：配備用バージョンへプラグインを組み込む際は、適切なライセンス番号が必要です。

シリアル番号の管理

新しいバージョンのアプリケーションビルダでは、作成するアプリケーションへ組み込むシリアル番号を指定することができます。

注：クライアント/サーバの実行可能なアプリケーションのライセンス管理は4D Sercerでも同様です。ライセンス番号あるいはエクスパンション番号を入力することで、サーバマシン上で「ライセンスの更新…」ダイアログボックスを用いて実行されます。

「ライセンス」タブをクリックして、マシン上の利用可能なライセンス一覧を表示します。



デフォルトでは、リストは空欄になっています。生成されたアプリケーションで使用される各4D Runtime Volume Licenseと同様に4D Developer Editionのライセンスを追加してください。ほかのDeveloper Editionのライセンスを追加し、現在使われているライセンスと関連づけることができます。

このリストには、配備用のライセンス番号だけが表示されます。ライセンス番号が失効すると、リスト上に臨時の番号が赤色で表示され、選択できなくなります。

番号の削除や追加を行うには、このウインドウの下側にある「+」ボタンおよび「-」ボタンを使用します。

「+」ボタンをクリックすると、ファイルオープン用のダイアログボックスが表示されます。ここで、配布用ライセンスを格納するシリアルファイルを指定しなければなりません。このファイル（HTML形式）は、Developer Editionおよび4D Runtime Volume Licensesの購入時に生成または更新されています。

注：ライセンス番号登録に関する詳細は、4th Dimension の『インストールガイド』を参照してください。

ファイルを選択した後、そのファイルに納められたライセンスの特性がリストに表示されます。つまり、製品番号やエクステンション番号、有効日付、関連する Developer Edition の番号（マスター欄）、およびその保存場所が示されます。このライセンスが無効であれば、メッセージにより警告されます。

有効なファイルを好きな数だけ指定することができます。実行形式アプリケーションの作成時に、4th Dimension は利用可能なライセンスのうち最適なものを使用します。

アプリケーションの構築後、新しい配布用ライセンスファイルは「Licenses」フォルダへ自動的に納められ、実行形式アプリケーションと同じ階層上（Windows）またはパッケージ内（Mac OS）に配置されます。

クライアントアプリケーションの自動更新

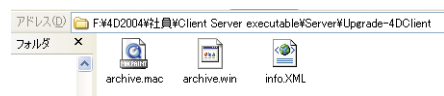
アプリケーションビルダの「クライアント/サーバ」ページにある「クライアントアプリケーションの自動更新を許可する」オプションを選択すると、作成したクライアント/サーバアプリケーションのクライアントパートは、構築済アプリケーションのバージョンが新しくなるたびに、ネットワーク経由で自動的に更新されます。

このシステムにより、開発者が各クライアントマシンにアップデートを手動でインストールする必要がなくなります。

アップデート通知の表示

クライアントアプリケーション更新の通知は、サーバアプリケーションの更新に続いて、実質上自動的に行われます。

アプリケーションビルダによりクライアント/サーバアプリケーションの新バージョンが作成されると、新しいクライアントパートが圧縮ファイルの形で「Server」フォルダにコピーされます。「アプリケーションビルド」ダイアログボックスで Windows 版と Mac OS 版の 4D Client を選択した場合（前述の「アプリケーションパラメータを設定する」の節を参照）、それぞれのプラットフォーム用の更新ファイルが利用できます。



クライアントアプリケーション更新の通知を起動するには、古いバージョンのサーバアプリケーションを新しいものに置き換えてから、それを実行します。その後の処理は自動的に行われます。

クライアント側で、“古い”クライアントアプリケーションが更新後のサーバアプリケーションへ接続しようとする、クライアントマシンにダイアログボックスが表示され、新しいバージョンが利用可能であることを知らせます。ユーザは、使用中のバージョンを更新するか、またはダイアログボックスをキャンセルすることができます。

■ ユーザが「OK」をクリックすると、ネットワーク経由でクライアントマシンに新しいバージョンがダウンロードされます。この際、進捗バーが表示されます。ダウンロードが完了すると、古いクライアントアプリケーションは終了し、新しいバージョンが起動してサーバーに接続します。この後、古いバージョンのアプリケーションは、マシンのごみ箱へ配置されます。

■ ユーザが「キャンセル」をクリックすると、更新はキャンセルされます。古いバージョンのクライアントアプリケーションのバージョンが、サーバが許容する範囲内（次の節を参照）ではない場合、アプリケーションは終了し接続できません。許容範囲であれば（デフォルト）、接続が行われます。

強制的に自動更新を行う

状況によって、クライアントアプリケーションがアップデートのダウンロードをキャンセルできないように設定しておきたい場合があります。例えば、4D Serverと4D Clientアプリケーションの新しいバージョンを使用する場合は、クライアントアプリケーションの新しいバージョンを必ず各クライアントマシン上にインストールしておかなければなりません。

強制的に更新を行うには、クライアントアプリケーションの現行バージョン番号（X-1より前の番号）をサーバアプリケーションに対応するバージョン番号範囲から除外します。この場合、更新メカニズムによって、未更新のクライアントアプリケーションの接続が許可されません。このパラメータは、XMLキーを使用してアプリケーションプロジェクトに設定します。

また、アプリケーションのバージョン番号（3桁の数値を使用）もXMLキーを使用して設定します。

XMLアプリケーションプロジェクトに関する詳細は、前述の「ダイアログボックスとビルドアプリケーションプロジェクト」の節、ならびに『4D XMLキー』マニュアルを参照してください。

Web サービス

4th Dimension 2004では、クライアント側とサーバ側の双方に関してWebサービス管理が強化されました。

- サーバ側では、4th Dimension はDOCモードでメソッドをWebサービスとして公開できます。
- クライアント側では、特に複合型の処理を行うことにより、4th Dimension 2004においてDOCモードで公開されたWebサービスをより簡単に利用することができます。

DOCモードでの公開（サーバ）

4th Dimension 2004では、メソッドをDOCモードのWebサービスとして公開できるようになります。特定のクライアントアプリケーションでは、この公開モードが要求されます。

DOC公開管理は4th Dimensionにより処理され、ユーザが気付くことは全くありません。4th Dimension 2003においてRPCモードのWebサービスとして公開されたメソッドは、そのまま4th Dimension 2004のDOCモードで公開することができます。さらに開発を行う必要はありません。

DOCモードでWebサービスを公開するには、WSDLのURL「<http://ServerAddress/4DWSDL/DOC>」を使用してください。

4th Dimensionサーバは、DOCモードでのリクエスト処理と応答を担当します。DOCリクエストは、「<http://サーバアドレス/4DSOAP/DOC>」（今までの<http://サーバアドレス/4DSOAP>ではなく）へ自動的に送信されます。

注：今までと同じように、「<http://サーバアドレス/4DWSDL>」というURLを使用し、4th DimensionによりRPCモードで公開されたWebサービスへアクセスすることができます。

SOAPライセンス

今後、4th Dimension 2004でWebサービスを公開する際は、特定のライセンスが必要になります。以前のバージョンの4Dでは、すべてのWebライセンスが必要でした。

Is license available関数を使用して、このライセンスが利用可能であるかどうかを調べるために、「Is license available」テーマに新しく2つの定数が追加されました。

定数	タイプ	値
4D SOAP License	倍長整数	808465464
4D Client SOAP License	倍長整数	808465465

Webサービスの利用（クライアント）

4th Dimension 2004では、新しく2つの機能が追加され、Webサービス（クライアント側）へよりアクセスしやすくなります。

- DOCモードで公開されたWebサービスのサポートが拡張されました。
- サーバへ送信されるSOAPリクエストのオプションを設定できるようになります。

DOCモードのサポート拡張

4th Dimension 2004では、DOCモードで公開されたWebサービスをより利用しやすくなります。Webサービスウィザードが複合型の処理をすべて行い、それを4Dコードへ変換してプロキシメソッドに納めます。

ウィザードを使用して、DOCモードにおけるWebサービスのプロキシメソッドを生成すると、そのリクエストの全XMLデータが手際よく解析されていきます。

XMLツリーを管理するコマンドを使用すると、Webサービスウィザードは複合型のXMLデータから利用可能なデータを取り出そうとします。次の例題は、DOCモードのWebサービスに関し、4th Dimension 2004で生成されたプロキシメソッドを示しています。

```

...
, proxy_WhoIs
, www.esynaps.com/WebServices/WhoIsService.asmx?WSDL
, メソッドのソースコードは4D SOAPウィザードによって自動的生成されます。
...

C_TEXT(1)
C_TEXT(2)

C_STRING(18:$root)
C_STRING(18:$subelem)
C_TEXT($namespace)
$namespace="http://tempuri.org/"
$root:=DOM Create XML Ref("WhoIs":$namespace)

$selelem:=DOM Create XML element($root;"/WhoIs/DomainName")
DOM SET XML ELEMENT VALUE($selelem;1)

SET WEB SERVICE PARAMETER("XMLIn":$root)
CALL WEB SERVICE("http://www.esynaps.com/WebServices/WhoIsService.asmx";"http://tempuri.org/WhoIs";"http://tempuri.org/";Web_Service_Manual_)

If (0=1)
  C_BLOB($blob)
  C_STRING(18:$resroot)
  C_STRING(18:$ressubelem)
  GET WEB SERVICE RESULT($blob;"XMLOut";*)
  $resroot:=DOM Parse XML variable($blob)
  $ressubelem:=DOM Find XML element($resroot;"/WhoIsResponse/WhoIsResult")
  DOM GET XML ELEMENT VALUE($ressubelem;2)
  DOM CLOSE XML ($resroot)
End if
DOM CLOSE XML ($root)

```

新しいXMLコマンドの
使用例

使用しているデータベースからプロキシメソッドを呼び出す方法は、以前のバージョンの4th Dimensionとまったく同じです。

注：Webサービスウィザードが完全に処理できるのは、配列および一つの深さの複合型データ（SOAPリクエストでは一つの階層レベル）だけです。

これより複雑な項目がリクエストに存在する場合、ウィザードはWSDLを発見するとメソッド名の横にフラグを表示し、これを示します。このタイプのWebサービスを取り扱う場合、一般的に開発者側で独自に解析処理を行う必要があります。

リクエストにオプションを設定する

4th Dimension 2004では、サーバへ送信するリクエストにオプションを設定することができます。次のオプションを指定できます。

- SOAP リクエストのヘッダ
- HTTP 接続のタイムアウト
- 処理される SOAP のバージョン

これらのオプションは、**SET WEB SERVICE OPTION** コマンドを使用して処理することができます。

はじめに

4th Dimension 2004には、データベースのフルバックアップおよび復元用モジュールが組み込まれています。

このモジュールは、スタンドアロン版の4th Dimensionならびに4D Serverで動作します。このモジュールにより、現在使用しているデータベースを終了せずに、そのバックアップを作成することができます。各バックアップには、ストラクチャファイル、データファイル（およびすべてのセグメント）、任意の追加ファイルやフォルダが含まれます。まず初めに、アプリケーションの「環境設定」でこれらのパラメータを設定します。

これらのバックアップの開始は、手動で、またはユーザ操作を必要とせずに一定間隔で自動的に行うことができます。特定のランゲージコマンドや新しいデータベースメソッドを使用すると、独自のインタフェースにバックアップ機能を統合することができます。

損傷したデータベースが開かれると、データベースは自動的に復元されます。

また、統合バックアップモジュールではログファイルを利用することができます。このファイルは、データに対して行われたあらゆる操作の記録を保管し、2回のバックアップ間の全体的な安全性も保証します。使用中のデータベースで問題が発生した場合、データファイルから失われたあらゆる操作は、次回そのデータベースをオープンすると自動的に再統合されます。特定のウインドウを使用すると、いつでもログファイルの内容を表示することができます。

統合されたバックアップモジュールを使用すると、次の事柄が実現します。

- 随時、データベースファイルのフルバックアップを開始する（ストラクチャファイル、ログファイル、同封ファイルなど）。
- 定期的な自動バックアップを導入する（時間単位、日単位、週単位、月単位）。
- 上級バックアップパラメータを設定する（セットの数、ファイルの圧縮、復元後の再起動オプションなど）。
- 障害が発生した場合に、データベースと同封ファイルを自動的に復元する。

- データベースのデータに関するバックトラッキング操作（処理の取り消し）を実行する。

バックアップの管理

バックアップ機能は、新しい4th Dimensionモジュールの主要な機能です。データベースのバックアップとは、一定の時点においてデータベースならびに必要となるすべてのファイルのコピーを作成することです。このコピーはバックアップファイル内に置かれます。

カレントデータベースを損傷するような障害が発生した場合には、4th Dimensionを使用してこれらのバックアップファイルを開くことができます。これにより、データベースは以前の状態に復元されます（コピーが作成された時点）。

バックアップの実行

各バックアップは、アプリケーションの「環境設定」で設定されたパラメータを考慮しながら実行されます。

「環境設定」を使用して、バックアップの各項目に関する設定を行います。

- バックアップに納めるファイル（データやログ、ストラクチャ、ユーザストラクチャ、同封ファイル）
- バックアップファイルの保管場所（主にバックアップファイルとログファイル）
- ログファイルの管理
- バックアップスケジュール
- バックアップオプション：バックアップセットの数およびローテーション、トランザクションやアクティブなインデックス操作の処理方法、セグメント化、圧縮、バックアップファイルの整合性
- 自動復元オプションの設定

これらのパラメータは、標準的な使用方法に合わせたデフォルト値に設定されています。これらの値の変更は任意です。これらのパラメータに関する詳細は、後述の「バックアップファイルの設定」ならびに「バックアップ設定」の節を参照してください。

バックアップの開始

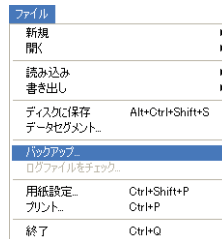
4th Dimension 2004 では、3通りの方法でバックアップを開始することができます。

- 手動による方法：4th Dimension の「ファイル」メニュー（「ユーザ」モード）や 4D Server の「バックアップ...」コマンドを使用する。
- 自動的に行う方法：アプリケーションの「環境設定」で設定されるスケジューラを使用する。
- プログラムによる方法：**BACKUP** コマンドを使用する。

4D Server：バックアップは、**BACKUP** コマンドを呼び出すメソッドを使用して 4D Client から手動で開始することができます。いかなる場合でも、このコマンドはサーバ上で実行されます。

▼ 手動バックアップを実行するには、次の手順に従ってください。

- 1 4th Dimension の「ファイル」メニュー（「ユーザ」モード）または 4D Server において、「バックアップ...」コマンドを選択する。



バックアップウィンドウが表示されます。

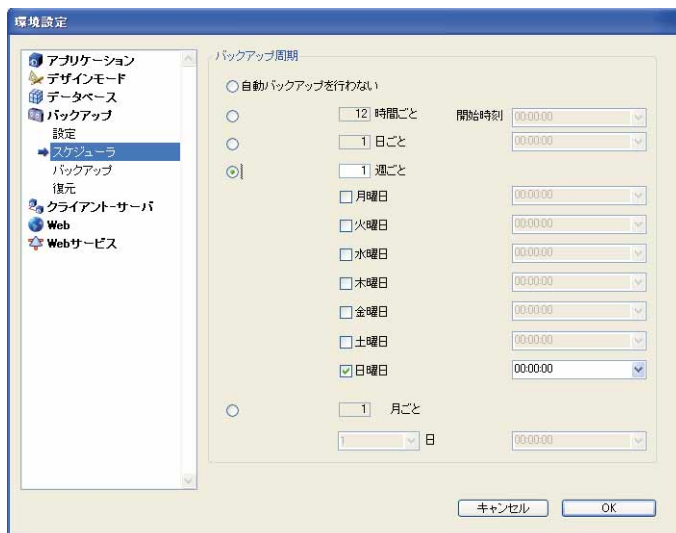


「設定」ボタンにより、バックアップの一般設定が表示されます（後述の「バックアップファイルの設定」の節を参照）。「キャンセル」ボタンはバックアップをキャンセルします。

- 2 「バックアップ」ボタンをクリックし、現在のパラメータを用いてバックアップを開始する。

▼ 定期的な自動バックアップを実行するには、次の手順に従ってください。

- 1 「バックアップ」環境設定の「スケジューラ」ページにおいて、バックアップのスケジュールを設定する。



ユーザが操作しなくても、バックアップはこのページで指定された時間に自動実行されます。

注：このダイアログボックスの使用に関する詳細は、後述の「定期的バックアップの設定」の節を参照してください。

- ▼ 4th Dimension ランゲージを使用して定期的バックアップを実行するには、次の手順に従ってください。

- 1 メソッドにおいてBACKUP コマンドを実行する。

現在のパラメータを用いてバックアップが開始します。

バックアッププロセスを処理するため、新しいデータベースメソッドである「On Backup Startup」および「On Backup Shutdown」を使用することができます。

この件に関する詳細は、後述の「バックアップ」の節を参照してください。

バックアップの実行

バックアップが開始されると、4th Dimension はバックアップの進捗状況を知らせるダイアログボックスを表示します。



「中止」ボタンをクリックして、いつでもバックアップを中断することができます（後述の「バックアップ中に問題が発生した場合」の節を参照）。

前回のバックアップの結果（成功または不成功）は、「環境設定」または4D Serverのメインウインドウにある「バックアップ／設定」ページの「最後のバックアップ情報」エリアに保管されます（後述の「最後のバックアップ情報」を参照）。また、この結果はデータベースのバックアップジャーナルにも記録されます（後述の「バックアップジャーナル」の節を参照）。

バックアップ中にデータベースへアクセスする

バックアップの実行中、次の場合はデータベースへアクセスすることができます。

- スタンドアロン版の4th Dimensionの場合：データベースの読み込み、書き込みはともにロックされ、すべてのプロセスが停止します。実行できるアクションはありません。
- 4D Serverの場合：データベースへの書き込みだけがロックされます。クライアントマシンはデータを表示することができます。クライアントマシンからサーバへ追加、削除、または変更のリクエストが送信されると、ウインドウが表示され、バックアップが終了するまで待機するよう求められます。

データベースの保存後、ウインドウが閉じられてアクションが実行されます。処理中のリクエストをキャンセルしてバックアップの終了を待たない場合は、「処理をキャンセル」ボタンをクリックしてください。

ただし、実行待機中のアクションがバックアップ前に開始されたメソッドのものである場合は、実行すべき残りの処理だけがキャンセルされてしまうので、このアクションはキャンセルしないでください。

しかも、メソッドの一部が実行済みであるため、データベースにおいて論理上の不整合が生じる可能性があります。

注：実行待機中のアクションがメソッドのものである場合に、ユーザが「処理をキャンセル」ボタンをクリックすると、4D Serverはエラー-9976を返します（データベースのバックアップが進行中のため、このコマンドは実行されない）。

バックアップ中に問題が発生した場合

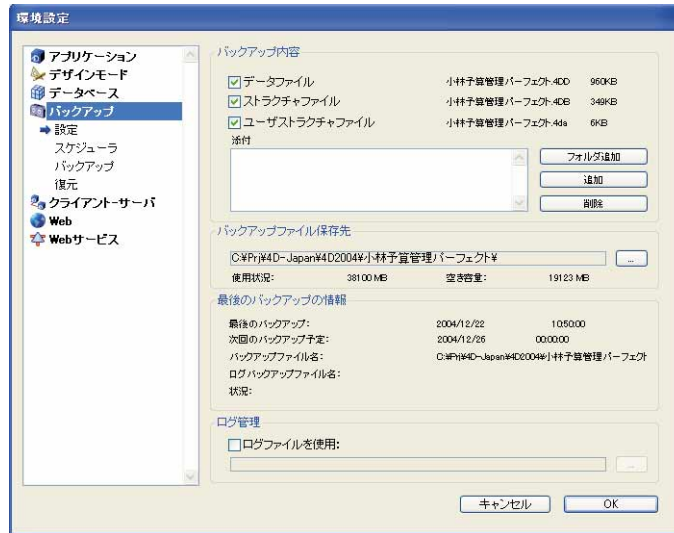
バックアップが正常に実行されない場合があります。バックアップが不成功に終わる原因としては、ユーザによる中断、同封ファイルが見つからない、保存先のディスクのトラブル、不完全なトランザクションなど、いくつか存在します。4th Dimension は原因に応じて問題に対処します。

すべての場合において、前回のバックアップのステータス（成功または不成功）は、「環境設定」の「バックアップ/設定」ページや4D Serverのウインドウ、バックアップログに表示されます（後述の「最後のバックアップ情報」を参照）。

- ユーザによる中断：進捗ダイアログボックスの「中止」ボタンを使用し、いつでもバックアップを中断することができます。この場合、各項目のコピーが中止されてエラーが生成されます。このエラーは「On Backup Shutdown」データベースメソッドで遮ることができます。
- 同封ファイルが見つからない：同封ファイルが見つからない場合、4th Dimension はバックアップを部分的に実行し（データベースファイルおよび使用可能な同封ファイルのバックアップ）、エラーを返します。
- バックアップ不可能（ディスクフル、書き込み防止、ディスクが見つからない、ディスク障害、不完全なトランザクション、定期的な自動バックアップの際にデータベースが起動されていない、など）
- 初回エラーの場合、4th Dimension は二回目のバックアップの実行を試みます。2回のバックアップ間の待機時間は、「環境設定」の「バックアップ」ページで指定します（後述の「バックアップ失敗時」を参照）。
- 再試行に失敗した場合、システムの警告ダイアログボックスが表示されてエラーが生成されます。このエラーは「On Backup Shutdown」データベースメソッドで遮ることができます。
- 不完全なトランザクションやインデックス付けのために中断されたバックアップは、特定のメカニズムを用いて処理されます（後述の「トランザクションまたはインデックス操作中の時」の節を参照）。

バックアップファイルの設定

アプリケーションの「環境設定」の「設定」ページでは、バックアップファイルとその保存場所、ならびにログファイルの保存場所を設定することができます。また、前回のバックアップに関する情報も提供されます。



4D Server：これらのパラメータは、4D Server マシン上でのみ設定することができます。

これらのパラメータは、4th Dimension アプリケーションによって開かれたデータベースごとに異なります。

バックアップの内容

このエリアを使用し、次回バックアップ時にコピーするファイルやフォルダを指定することができます。

このエリアの上部には、4th Dimension データベースの各ファイルが記載され、現在のサイズがそれぞれ示されます。対応するオプションをチェックし、各ファイルがバックアップに組み込まれるように設定しなくてはなりません。グレー表示されたオプションは、対応するファイルがデータベースで使用できないことを表わします。バックアップのスケジュール、戦略上の利点、サイズなどの条件に応じて、目的の4th Dimension ファイルを選択することができます。ファイルは必要ありません。

エリアの下部分には、バックアップ内の同封ファイルのフルアクセスパスが一覧表示されます。

■ データファイル：データベースのデータファイル

データベースが複数のセグメントに分かれている場合、1番目のセグメントの名前とサイズだけが表示されます。もちろん、すべてのセグメントがバックアップされます。

このオプションを選択すると、データベースのカレントログファイルが存在すれば、それがデータと一緒にバックアップされます。バックアップにより、カレントログファイルのクローズとバックアップが行われ、その後、新しいログファイルが作成されます。これによりログファイルのサイズが大きくなりすぎることを避けられます。

ログファイルに関する詳細は、後述の「ログファイルの管理」の節を参照してください。

■ **ストラクチャファイル：データベースのストラクチャファイル**

データベースがコンパイルされて4D Engineが組み込まれている場合、このオプションを使用して「.exe」ファイル（Windows）やパッケージ（Mac OS）をバックアップすることができます。

■ **ユーザストラクチャファイル：カスタマイズされたユーザフォームを含むデータベースのユーザストラクチャです。このファイルに関する詳細は、後述の「ユーザフォーム」の節を参照してください。**

■ **同封ファイル：このエリアを使用して、データベースと一緒にバックアップされる一連のファイルやフォルダを指定することができます。これらのファイルはいずれのタイプでも構いません（ドキュメント、プラグインテンプレート、ラベル、レポート、ピクチャなど）。**

個々のファイルまたは、全内容がバックアップされるフォルダのいずれかを設定できます。同封する各項目は、そのフルアクセスパスとともに「同封」エリアに一覧表示されます。

■ **フォルダの追加：このボタンをクリックすると、4th Dimensionよりダイアログボックスが表示され、バックアップに追加するフォルダを選択することができます。復元の際に、フォルダはその内部ストラクチャとともに復旧されます。データベースファイルを含むフォルダを除き、マシンに接続されたあらゆるフォルダやボリュームを選択することができます。**

■ **追加：このボタンをクリックすると、4th Dimensionよりダイアログボックスが表示され、バックアップに追加するファイルを選択することができます。データベースファイルを同封ファイルとして選ぶことはできません。**

■ **削除：このボタンにより、選択したファイルを同封ファイル一覧から削除することができます。**

注：バックアップを実行する際に、いくつかのファイルが利用できない可能性もあります（名前やアクセスパスの変更、ボリュームに空きがないなど）。この場合、不足ファイル抜きでバックアップが実行され、エラーが生成されます。このエラーは「On Backup Shutdown」データベースメソッドで遮ることができます。また、エラーはバックアップログにも残されます。

バックアップファイル保存先

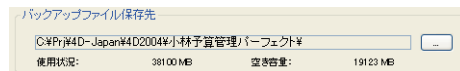
このエリアには、バックアップファイルとログファイル（該当する場合）を保存する場所を指定します。

4th Dimension は、バックアップファイルおよびバックアップログファイルという2種類のバックアップファイルを生成します。バックアップファイルは、次の2つの理由において特別です。

- バックアップファイルには複数のファイルを納めることができます（ストラクチャ、データ、同封ファイルなど）。
- これらのファイルは、内部検証メカニズムを用いて保護されています（上級バックアップ設定を用いて指定）。

デフォルトとして、4th Dimension はこれらのファイルをデータベースのデータファイルと同じ階層に保存します。データベースやバックアップが格納されているドライブ上でディスク障害が発生した場合に、データ損失の危険性を抑えるため、別のディスクボリューム上の場所を設定することを強くお勧めします。

これらのファイルの保管場所を変更するには、「...」ボタンをクリックします。



選択用のダイアログボックスが表示され、バックアップを格納するフォルダやディスクを選択することができます。「使用済領域」と「空き領域」エリアが自動的に更新され、選択したフォルダのディスク上の残存ディスク領域が示されます。

すべてのバックアップに対して十分な空き領域があることを確認してください。ディスク容量不足のためバックアップが不成功に終わると、エラーが生成されます。このエラーは「On Backup Shutdown」データベースメソッドで遮ることができます。また、エラーは情報エリアやバックアップログに示されます。

バックアップファイル名

4th Dimension は、特定の命名方式を用いてバックアップファイルの名前を設定し、自動復元機能はこの命名方式に基づいて行われます。この命名方式を変更することはできません。

- 標準的なバックアップには、“データベース名[xxxx].4BK”という名前が付けられます。「データベース名」はデータベースのデータファイル名であり、「xxxx」はバックアップの回数です。例えば、請求書データベースの26回目のバックアップには、請求書[0026].4BKという名前が付けられます。

バックアップがセグメント化されている場合、4th Dimensionは“-xxxx”のようにセグメント番号を付加します。例えば、請求書データベースの26回目のバックアップの3番目のセグメントには、請求書[0026-0003].4BKという名前が付けられます。

セグメントに関する詳細は、後述の「アーカイブ」の節を参照してください。

- ログファイルのバックアップには、“ログ名[xxxx].4BL”という名前が付けられます。「ログ名」はログファイル名であり、「xxxx」はバックアップの回数です（0から開始）。例えば、ログファイル「Log」の13回目のバックアップには、Log[0012].4BLという名前が付けられます。

ログファイルのバックアップがセグメント化されている場合、4th Dimensionは“-xxxx”のようにセグメント番号を付加します。例えば、ログファイル「Log」の13回目のバックアップの2番目のセグメントには、Log[0012-0002].4BLという名前が付けられません。

セグメントに関する詳細は、後述の「アーカイブ」の節を参照してください。

ログファイルのバックアップは0から始まり、データベースファイルのバックアップは1から始まる点に注意してください。データベースファイルの最初のバックアップ（例えば、backup[0001].4BK）の場合、ログファイルのバックアップの名前はlog[0000]になります。ログファイルは、“空白”の状態から開始したデータファイルへの変更を表わし、空のデータファイルにのみ統合されます。よって、例えばlog[0025].4BLという名前のログバックアップは、「25回目と26回目のデータベースバックアップの間に実行された操作に対応するログファイルの、26回目のバックアップ」であるものと解釈しなくてはなりません。したがって、log[0025].4BLはバックアップbackup[0025].4BKに対応します。

注：バックアップの番号付け範囲は次の通りです。

- ・バックアップ：1～9999
- ・ログファイルのバックアップ：0～9998
- ・セグメント：1～9999

最後のバックアップ情報

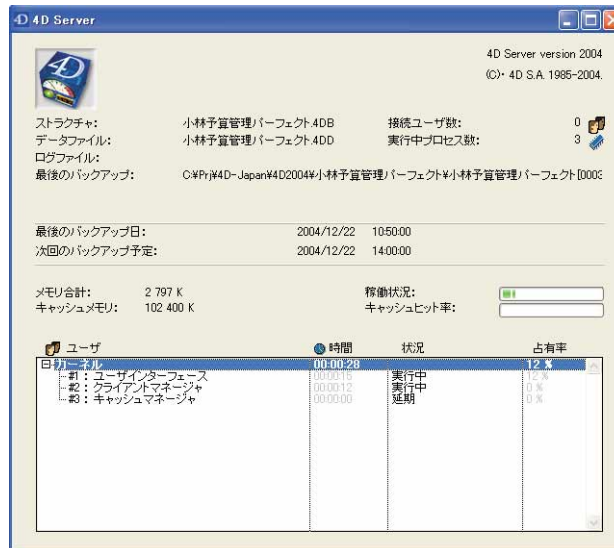
「最後のバックアップ情報」には、データベースの前のバックアップに関する情報が提供されます。少なくとも1回のバックアップが実行されていれば、この情報が提示されません。

- 日付と時間：前回バックアップの日付と時間。
- ステータス：このエリアには、前回バックアップのエラーコードおよびそのコードの説明が表示されます。バックアップが正常に実行された場合、コード0（バックアップ成功）が表示されます。

定期的バックアップに関し、このエリアを使用して前回のバックアップがスケジュール通りに行われたかどうかを検証することができます。

- **バックアップファイル名**：前回の主要なバックアップのアクセスパスとファイル名。
バックアップがセグメント化されている場合、1番目のセグメント名が表示されます。
- **ログバックアップファイル名**：前回のログファイルバックアップ（該当する場合）のアクセスパスとファイル名。
- **次のバックアップ予定**：次回バックアップの日付と時間。バックアップスケジュールが導入されている場合、このエリアに情報が提示されます。

4D Serverを使用している場合、サーバのメインウィンドウにも、この情報が表示されます。

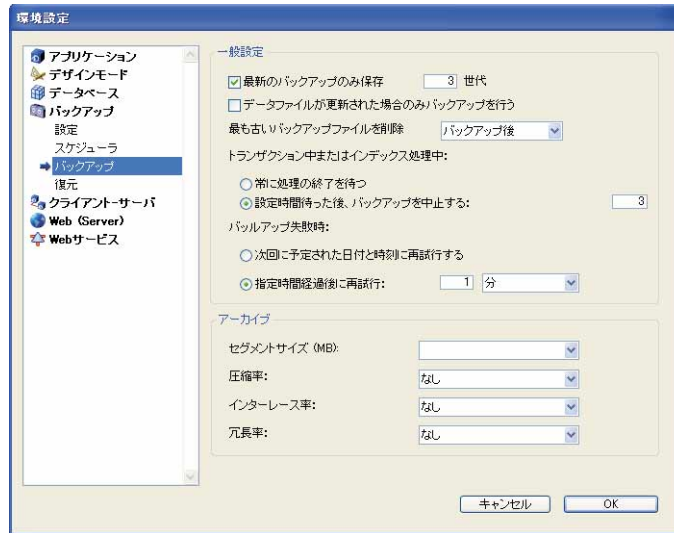


注：ログファイル管理に関連するパラメータについては、後述の「ログファイルの管理」の節で説明しています。

バックアップ設定

環境設定と同様に、「バックアップ」設定は各バックアップごとに使用されます。また、これら設定の変更はすべて任意です。それぞれのデフォルト値は、バックアップ機能の標準的な使用方法に合わせています。

バックアップ設定は、アプリケーションの「環境設定」の「バックアップ」ページで指定します。



4D Server：バックアップ設定は、4D Server マシンからのみ設定することができます。

これらの設定は、4th Dimension アプリケーションで開かれたデータベースごとに異なります。

最新のバックアップのみ保存する

このパラメータを使用し、最も古いバックアップファイルの削除に対して使用するメカニズムの起動や設定を行うことができます。このメカニズムにより、最新のバックアップファイルのうち、特定数をバックアップディスク上に残しておくことができます。この後、新しくバックアップを行うたびに、最も古いファイルが削除されます。これにより、ディスクドライブが一杯になる危険性を回避します。

この機能は次の手順で動作します。現在のバックアップが完了すると、4th Dimension は、バックアップするアーカイブと同じ場所に最も古いアーカイブが存在し、しかも同じ名前である場合にそれを削除します。スペース節約のため、バックアップ前に一番古いアーカイブを削除するよう要求することができます。

例えば、セット数を3に設定すると、最初の3回のバックアップにより MyBase-0001、MyBase-0002、MyBase-0003 というアーカイブがそれぞれ作成されます。4回目のバックアップの際には、MyBase-0004 というアーカイブが作成され、MyBase-0001 が削除されません。

バックアップ設定を行ったディスク上の場所から開始する場合、次の式を用いて使用できる最大のバックアップセット数を決定することができます。

$$\text{最大セット数} = \frac{\text{バックアップに使用できる容量}}{\text{バックアップの最大サイズ}} - 1 \text{ セット}$$

デフォルトとして、4th Dimension は最初に現在のバックアップを実行し、その後、最も古いアーカイブをディスクから削除するため、最大セット数から1を減じなければなりません。この動作は変更可能です（後述の「データファイルが修正された時にバックアップ」の節を参照）。

デフォルトでは、セットを削除するメカニズムが有効であり、4th Dimension は3セットのバックアップを保管します。

このメカニズムを無効にするには、このオプションの選択を解除します。

注：このパラメータは、データベースのバックアップセットとログファイルのバックアップセットの双方に影響します。

データファイルが更新された場合のみにバックアップを行う

このオプションを選択すると、4th Dimension は、前回のバックアップ以降にデータベースのデータの追加、変更、削除が行われた場合にのみ、定期的バックアップを開始します。それ以外の場合、定期的バックアップはキャンセルされ、次回の定期バックアップまで持ち越されます。エラーは返されませんが、ログファイルにはバックアップが延期されたことが記されます。

また、このオプションにより、大半が閲覧に費やされるデータベースバックアップのマシン時間を節約することができます。このオプションを有効にしても、ストラクチャファイルや同封ファイルへの修正はまったく考慮されない点に注意してください。

注：

- ・定期的バックアップに関する詳細は、後述の「定期的バックアップの設定」の節を参照してください。
- ・このオプションを選択して手動でバックアップを開始すると、4th Dimension はダイアログボックスを表示して、データファイルが変更されていないことを知らせます。ユーザは、このダイアログボックスで処理の確定やキャンセルを行うことができます。
- ・このパラメータは、データベースならびにログファイルのバックアップの双方に影響します。

最も古いバックアップファイルの削除

このオプションは、「最新のバックアップのみ保存する」オプションが選択されている場合にのみ使用できます。このオプションを使用すると、4th Dimensionが最も古いアーカイブを削除してからバックアップを開始するか（「前」オプション）、またはバックアップが終了してから削除を行うか（「後」オプション）を指定することができます。このメカニズムを機能させるためには、最も古いアーカイブのリネームや移動を行ってはいけません。

「バックアップ前」オプションを使用するとスペースを節約することができます。許可されたバックアップ数から1を減らす必要はなくなります。これにより、例えば5GBのディスク上に2GBのアーカイブを2つ保存することができます。また、注意すべきなのは、バックアップが中断された場合、古いバックアップは利用できず（事前に削除されているため）、現在のバックアップは未完了なのでこれも利用できないという点です。バックアップ中はデータベースへの修正が行えないため、有効なバックアップが存在しない場合でも、データファイルは変更されないことが保証されます。システムを再起動して不完全なアーカイブを破棄したら、即座にバックアップを新たに実行することができます。

「後」オプションは、補足的な安全対策ですが、バックアップするハードディスクのスペースをより多く必要とします。

トランザクション中またはインデックス処理中

トランザクションならびにインデックス操作は、その操作が完了しない限り、データベースのバックアップを妨げる重大な操作です。4th Dimensionや4D Serverはバックアップを開始する前に（開始方法に関係なく）、データベースの検証を行います。プログラムは、実行中である重要な処理を検出すると、バックアップ手続きを一時中断します。

このパラメータにより、この待機時間を指定することができます。

- 常に処理の終了を待つ：バックアップは、これらの処理が完了するまで中断されます。処理の完了後、バックアップが即座に開始します。これにより、バックアップは必ず実行されますが、バックアップの開始時に厳密な検証が行えません。
- 設定時間待った後、バックアップを中止する：バックアップは一定時間中断され（デフォルトでは3分間）、トランザクションやインデックス処理が完了するための猶予が与えられます。この待機時間が過ぎても重要な処理がまだ実行中であれば、バックアップは失敗したものとみなされてバックアップログにエラーが記録されます。この後、バックアップが失敗した場合の処理として定義されたメカニズムが適用されます（後述の節を参照）。

注：データベースが起動している時のトランザクションのオープン、あるいは、データベースを終了する際に確定またはキャンセルされただけのトランザクションのオープンは避けてください。また、トランザクション処理中に、ユーザに対して入力や確定用のダイアログボックスを表示しないようにしてください。後者の場合に、ダイアログボックスが確定されないまま、その時点でバックアップが開始すると、データベースではそのダイアログボックスが確定されるまで新しいトランザクションを実行できません。

バックアップ失敗時

このオプションにより、不成功のバックアップ処理に対して用いるメカニズムを定義することができます。

注：

- ・すべてのトラブルがバックアップ失敗に結びつく訳ではない点に注意してください（前述の「バックアップ中に問題が発生した場合」の節を参照）。
- ・定期的自動バックアップの実行予定時間にデータベースが起動されていない場合、4th Dimension はバックアップが失敗したものとみなします（「定期的バックアップ」の節を参照）。

バックアップを実行できない場合、4th Dimension は再度バックアップを試みます。2つのオプションが用意されています。

- 次回に予定された日付と時刻に再試行する：このオプションは、定期的な自動バックアップ機能を用いて作業している場合にのみ意味を持ちます（後述の「定期的バックアップの設定」の節を参照）。このオプションは、不成功のバックアップをキャンセルし、エラーを生成します。
- 指定時間経過後に再試行：このオプションを選択すると、待機時間の後にバックアップが新たに実行されます。このメカニズムにより、バックアップの妨げとなる一定の状況を見越しておくことができます。待機時間は、対応するメニューを使用して時間単位または分単位で設定することができます。再試行も不成功であった場合はエラーが生成され、不成功の結果が前回のバックアップのステータスエリアとバックアップログファイルに記録されます。

アーカイブ

このエリアを使用して、アーカイブの生成オプションを設定することができます。これらのオプションは、主なバックアップファイルとログファイルのバックアップファイルに対して適用されます。

■ セグメントサイズ (MB)

4th Dimensionでは、アーカイブをセグメント化することができます。セグメント化とは、アーカイブをより小さなサイズに分割するということです。例えば、この処理により、バックアップを複数の媒体に保存することができます (CD、ZIPなど)。復元作業中に、4th Dimensionは自動的に各セグメントを統合します。セグメントにはそれぞれ、MyDatabase[xxxx-yyyy].4BKという名前が付けられ、xxxxにはバックアップ回数、yyyyにはセグメント番号が設定されます。例えば、データベースMyDatabaseのバックアップの3つのセグメントは、それぞれMyDatabase[0006-0001].4BK、MyDatabase[0006-0002].4BK、MyDatabase[0006-0003].4BKという名前になります。

コンボボックスの「セグメントサイズ」メニューを使用して、バックアップの各セグメントのサイズをMB単位で設定することができます。事前に設定されているサイズからいずれかを選択するか、または0から2048までの具体的なサイズを入力することができます。このサイズに0を指定した場合、セグメント化は行われません (「なし」を指定した場合と同じ)。

■ 圧縮率

デフォルトとして、4th Dimensionはディスクスペースの節約に役立つように、バックアップを圧縮します。一方で、大容量のデータを処理する場合、ファイル圧縮フェーズによりバックアップ速度が著しく低下する可能性があります。

「圧縮率」オプションにより、ファイル圧縮を調節することができます。

■ なし：ファイル圧縮は適用されません。バックアップは高速に行われますが、アーカイブファイルはかなり大きくなります。

■ 速度優先 (デフォルト)：このオプションは、バックアップ速度とアーカイブサイズとの折衷策です。

■ 圧縮率優先：アーカイブに対して最大圧縮率が適用されます。アーカイブファイルが占めるディスク上の容量は可能な限り小さくなりますが、バックアップ速度が著しく低下します。

■ インターレース率とリダンダンシー率

4th Dimensionは、最適化 (インターレース) とセキュリティ (リダンダンシー) メカニズムに基づいた特定のアルゴリズムを使用し、アーカイブを生成します。これらのメカニズムは、ニーズに合わせて設定することができます。これら2つのオプションに関し、4th Dimensionはデフォルトの割合である「中」を提供します。また、これらオプションのメニューには、「低」、「高」、「なし」という割合が用意されています。

■ インターレース率：インターレースは、読み込み速度を向上するため、連続していないセクタにデータを保存します。しかし、保存フェーズの速度は遅くなります。

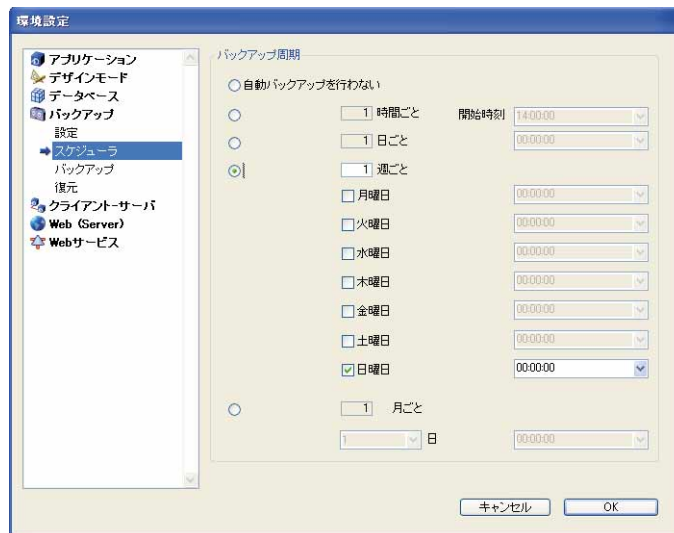
- **リダンダンシー率**：リダンダンシー（冗長度）は、同じ情報を複数回繰り返すことにより、ファイル内に保存されたデータを保護します。リダンダンシー率が高くなるほど、ファイルのセキュリティは強化されますが、保存速度は遅くなり、ファイルサイズは大きくなります。

定期的バックアップの設定

4th Dimension や 4D Server で開かれたデータベースのバックアップを自動化することができます（クライアントマシンがまったく接続していない場合でも）。そのためには、バックアップスケジュールを設定する必要があります（時間単位、日単位、週単位、月単位）。各セッションごとに、4th Dimension は「環境設定」に指定されたバックアップ設定を用いて、バックアップを自動的に開始します。

バックアップの予定時間にアプリケーションが起動していなかった場合、4th Dimension は次回起動した際、このバックアップが失敗したものとみなし、データベースの「環境設定」に指定された適切な設定を適用します。

バックアップ設定のスケジューラは、アプリケーションの「環境設定」の「スケジューラ」ページで定義します。



まず初めに、対応するラジオボタンをクリックして、スケジュールの単位（日、週など）を選択しなくてはなりません。その次に、バックアップを実行する時を指定します。

- **非自動バックアップ**：定期バックアップ機能は無効です。
- **時間ごと**：このオプションにより、時間単位でバックアップを予定することができます。1から24までの値を入力可能です。

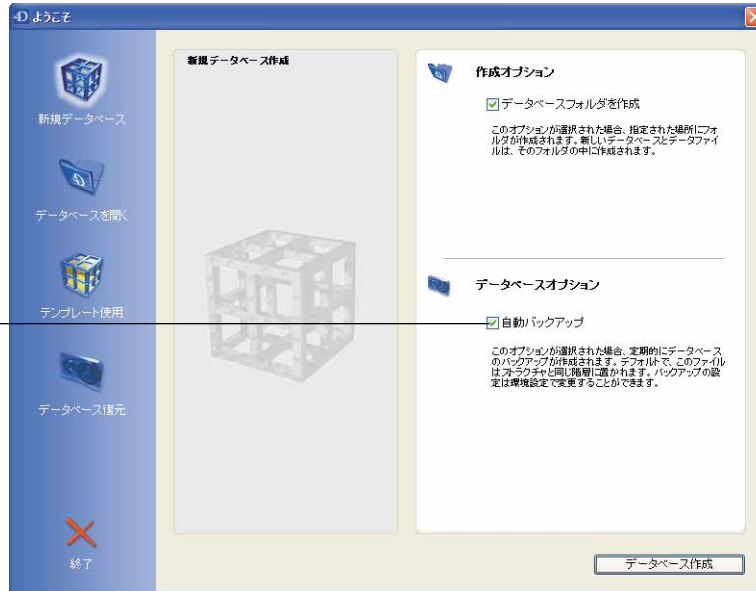
- **開始時刻**：時間毎のバックアップを最初に開始する時間を設定します。
- **日ごと**：このオプションにより、日単位でバックアップを予定することができます。毎日バックアップを実行したい場合は、1を入力します。このオプションを選択した場合、バックアップを開始する時間を入力しなくてはなりません。
- **週ごと**：このオプションにより、週単位でのバックアップを予定することができます。毎週バックアップを実行したい場合は、1を入力します。このオプションを選択した場合、バックアップを開始する曜日と時間を入力しなくてはなりません。必要であれば、複数の曜日を選択することができます。例えば、このオプションを使用し、毎週の水曜と金曜日の2回にバックアップを設定することができます。
- **月ごと**：このオプションにより、月単位でのバックアップを予定することができます。毎月バックアップを実行したい場合は、1を入力します。このオプションを選択した場合、バックアップを開始する日と時間を入力しなくてはなりません。

自動バックアップ

4th Dimension はデータベースの作成時に、デフォルトのバックアップ計画を提供します。バックアップ計画には一連のバックアップパラメータが用意され、最低限のデータベースセキュリティが保証されています。

データベースの作成後に、これらの設定を変更することができます。

データベースの新規作成時は、デフォルトとしてバックアップ計画が有効になっています。これを無効にするには、「自動バックアップ」オプションの選択を解除します。このオプションは、「ようこそ」ダイアログボックスの「作成」タブの右下にある展開ボタンをクリックすると表示されます。



デフォルトのバックアップ計画には、次の設定が含まれています。

- ログファイルの作成と使用（ファイル名は、データベース名.4dl）
- データベースストラクチャファイルと同じ階層にあるすべてのデータベースファイルのバックアップ（データ、ログ、ストラクチャ、ユーザストラクチャ）
- 毎日曜00:00:00時の自動バックアップ
- デフォルトの上級設定（最新のバックアップを3つ保管、圧縮率“中”など）
- すべての自動復元オプション

ログファイルの管理

継続的に使用されるデータベースには、変更ならびにレコードの追加や削除が常に行われています。データを定期的にバックアップすることは重要ですが、バックアップだけでは（予期しない障害の場合に）、前回のバックアップ以降に入力されたデータを回復することはできません。このニーズに応えるため、4th Dimensionは特定のツールであるログファイルを提供しています。このファイルを使用すると、データベースデータのセキュリティが常に保証されます。

さらに、4th Dimensionは絶えずメモリ上のデータキャッシュを用いて作業を行います。データベースのデータへの変更はすべて、ハードディスクへ書き込む前に、一時的にキャッシュへ保存されます。これにより、アプリケーションの処理速度が向上します。実際、メモリへのアクセスは、ハードディスクへのアクセスよりも高速に行われます。

キャッシュに保存したデータをディスクへ書き込む前に障害が発生した場合は、カレントログファイルを組み込んでデータベースを完全に復旧しなくてはなりません。

はじめに

4Dが生成するログファイルには、データベース上で行われたすべての操作が順次記録されています。したがって、ユーザが実行した各操作により、2つのアクションが同時に行われます。一つは、データベースに対するアクション（命令を正常に実行）、もう一つはログファイルに対するアクション（操作の説明を記録）です。ログファイルは個別に作成され、ユーザの作業を妨げたり作業速度を低下させることはありません。1つのデータベースでは、一度に1つのログファイルだけを使用して作業することができます。

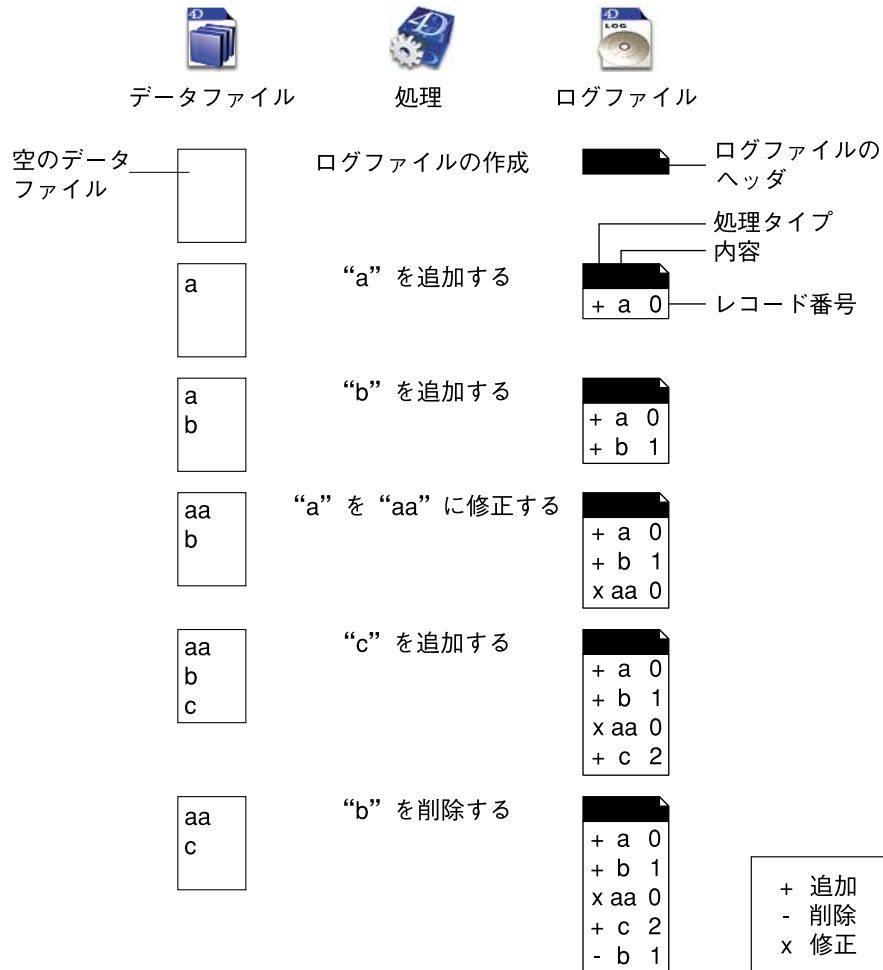
ログファイルには、4種類の操作が記録されます。

- レコードの追加
- レコードの変更
- レコードの削除
- トランザクション

トランザクションに関する注意：トランザクションは、同時に実行される一連の操作です。例えば、トランザクションで実行された銀行送金の場合、進行中の処理を妨げるようなリスクを冒さずに、同じオプションで一方の口座を借方勘定に記入し、もう一方の口座は貸方勘定に記入しなくてはなりません。この場合、すべての処理が実行されるか、またはまったく実行されないかのいずれかであり、操作を部分的に行うということは決してあり得ません。つまり、一方の口座が貸方勘定に記入されていなければ、もう一方の口座が借方勘定に記入されることはありません。4th Dimensionは、追加や変更、削除と同じようにトランザクションを単独の操作として解釈します。

4th Dimensionはログファイルを処理します。ログファイルは、データファイルに作用するあらゆる操作を区別なく盛り込み、ユーザが行った操作（スタンドアロン版の4th Dimensionや4D Client）や4Dメソッドによる処理、4Dプラグイン（4D Write、4D View、4D Drawなど）、Webブラウザ、4D Serverを用いた4D Openによる処理など、あらゆる操作を対象とします。

ログファイルの機能をまとめた図を次に示します。



カレントログファイルはカレントデータファイルと一緒に自動保存されます。このメカニズムには、異なる2つの利点があります。

■ ログファイルを保存するディスクの容量が一杯になる危険性を回避します。バックアップを実行しない場合、ログファイルは使用するにつれて徐々に大きくなり、いずれはディスクの空き容量をすべて使い果たしてしまいます。データファイルをバックアップするたびに、4th Dimensionや4D Serverはカレントログファイルをクローズして、即座に新規の空ファイルから開始するため、この危険を避けることができます。この後、古いログファイルはアーカイブに保存され、バックアップ設定メカニズムに従って最終的には破棄されます。

■ 後でデータベースの解析や補修を行うために、バックアップに対応するログファイルを保管します。ログファイルの統合は、それに対応するデータベースからのみ実行できます。

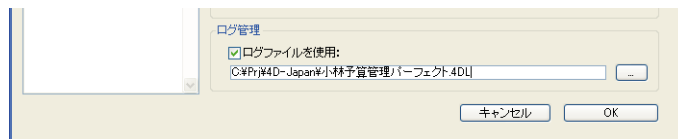
バックアップにログファイルを適切に統合するためには、バックアップとアーカイブ化されたログファイルを同時に保管することが重要です。

ログファイルの作成

デフォルトでは、4th Dimension 2004で作成されたあらゆるデータベースでログファイルが使用されます。このファイルの作成は、データベース作成用のダイアログボックス上にある「自動バックアップ」オプションと関連しています（前述の「自動バックアップ」の節を参照）。

ログファイルには「データファイル名.4DL」という名前が付けられ、データベースストラクチャを含むフォルダ内に置かれます。

データベースにおけるログファイルの使用状況は、いつでも調べることができます。ログファイルが使用されている場合は、「設定」ページの「使用ログファイル」オプションが選択されています（前述の「バックアップ設定」の節を参照）。



4D Serverでは、ログファイル名もサーバマシンのウィンドウ上に表示されます。

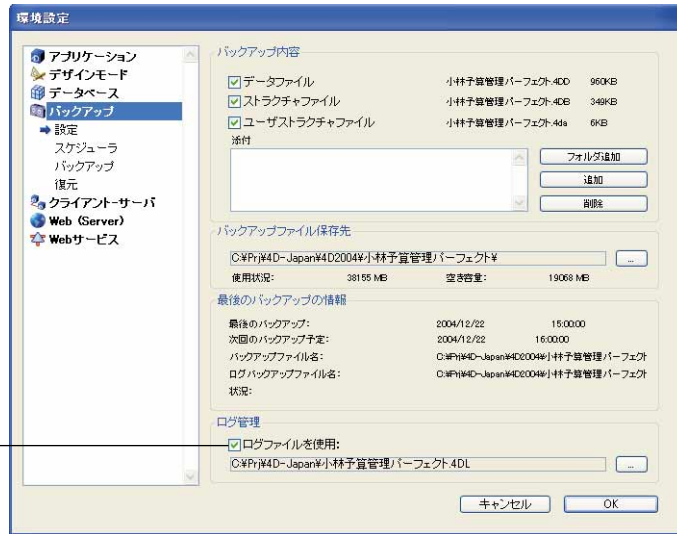


このオプションの選択を解除した場合、またはログファイルなしでデータベースを使用しており、ログファイルを用いたバックアップ計画を設定した場合は、ログファイルを作成する必要があります。

▼ ログファイルを作成するには、次の手順に従ってください。

- 1 アプリケーションの「環境設定」の「設定」ページ（「バックアップ」テーマ）において、「使用ログファイル」オプションを選択する。

「使用ログファイル」
オプション



互換性に関する注意：このオプションは、前バージョンの4th Dimensionの「ログファイルに必ず記録する」オプションに代わるものです。

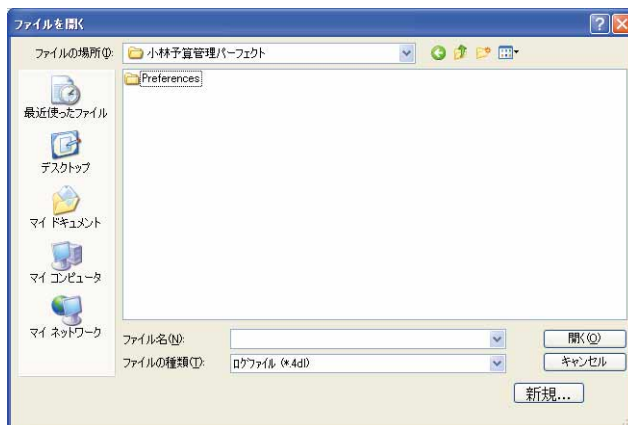
ログファイルを作成できるようにするには、データベースが次の条件のいずれかを満たさなくてはなりません。

- データファイルが空である。
- データベースのバックアップを実行した直後であり、データへの変更がまだ行われていない。

他のすべての場合において、プログラムはバックアップを実行するかどうかを尋ねる警告ダイアログボックスを表示します。このような予防措置が不可欠な理由は、あらゆる障害の発生後にデータベースを復元するため、ログファイルに記録された処理を統合するデータベースのコピーが必要となるためです。



データベースが上記の条件のいずれかを満たす場合、プログラムは標準の「ファイルを開く」ダイアログボックスまたは新しい「ログファイル」ダイアログボックスを表示します。



2 「新規...」 ボタンをクリックする。

標準の「ファイル保存」ダイアログボックスが表示されます。デフォルトとして、ログファイルには「データファイル名.4DL」という名前が付けられます。

3 デフォルトの名前を維持するか、または名前を変更した後、ファイルの保管場所を選択します。

少なくとも3つのハードドライブがある場合、データベースが保管されているディスク以外の場所にログファイルを保存することをお勧めします。これにより、データベースが保管されているハードドライブが破損した場合でも、ログファイルを呼び出すことができます。

4 「保存」 をクリックする。

開いたログファイルのアクセスパスと名前が「環境設定」ダイアログボックスの「使用ログファイル」エリアに表示されます。

この他に何も行わなくても、データに対して実行したすべての処理がこのファイルに記録され、その後データベースをオープンした際にこのファイルが利用されます。

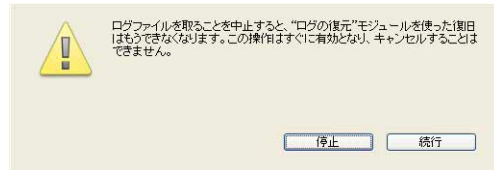
■ 新規データファイルを作成したら、別のログファイルを作成しなくてはなりません。

■ ログファイルに関連付けられていない（または、ログファイルが見当たらない場合）別のデータファイルをオープンした場合、他のログファイルを設定するか、または作成しなくてはなりません。別のログファイルを開くには、ログファイルのアクセスパス表示エリアの右側にある「...」 ボタンをクリックします。

ログファイルを中止する

カレントログファイルへの操作記録を中止したい場合、アプリケーションの「環境設定」の「設定」ページ（「バックアップ」テーマ）内の「使用ログファイル」オプションを選択解除します。

すると、4th Dimension は警告メッセージを表示して、この動作によりログファイルによるセキュリティが利用できなくなることを知らせます。



「停止」をクリックすると、カレントログファイルが即座にクローズされます（後から「環境設定」ダイアログボックスを確定する必要なし）。

カレントログファイルが大きすぎるため、それをクローズしたい場合は、データファイルのバックアップを実行してください。これにより、ログファイルのバックアップが作成されます。

ログファイルの解析

4th Dimension には、カレントログファイルの内容を表示する機能が用意されています。この機能は、データベースの運用を解析したり、エラーや不具合の原因となる操作を検出する際に役立ちます。クライアント/サーバモードにおけるデータベースの場合、この機能により各クライアントマシンで実行された処理を検証することができます。

この機能を使用して、データベースデータ上で実行されたすべての処理を遡ることができます。この機能については、後述の「処理の取り消し」の節で説明します。

カレントログファイルを表示するには、4th Dimension または 4D Client（サーバ側では、この機能を使用できません）の「ファイル」メニューから「ログファイルをチェック...」コマンドを選択します。

ファイル	
新規開く	▶
読み込み書き出し	▶
ディスクに保存 データセグメント...	Alt+Ctrl+Shift+S
バックアップ	
ログファイルをチェック...	
用紙設定... プリント...	Ctrl+Shift+P Ctrl+P
終了	Ctrl+Q

次のウィンドウが表示されます。



ログファイルに記録された各処理が行形式で表示されます。各カラムには、操作に関するさまざまな情報が提供されます。また、デフォルトのカラム表示を変更することができます。例えば、このウィンドウのコンテキストメニューを用いてフィールドの値を表示することができます（後述の「カラム表示の設定」の節を参照）。

この情報を利用して、各操作の実行箇所や状況を特定することができます。

- **アクション**：データに対して実行された操作のタイプ。このカラムには、追加、削除、変更、トランザクションのいずれかが納められます。詳細については、前述の「はじめに」の節を参照してください。
- **テーブル**：追加・削除・変更が行われたレコードが属すテーブル。
- **ユーザ**：操作を実行したユーザの名前。クライアント／サーバ環境では、クライアント側マシンの名前が表示されます。スタンドアロンの場合、ユーザのIDが表示されます。
4D Open を使用して操作が実行された場合、“4D Open” というラベルが表示されます。
- **日付と時間**：操作が実行された日付と時間。

カラム表示の設定

「ログファイル」ウィンドウのコンテキストメニューを使用して、カラム表示をカスタマイズすることができます。カレントウィンドウのセッションに関して、カラムの追加や削除、またはカラム内容の置き換えを実行することができます（ウィンドウが閉じられると、カスタマイズしたウィンドウの内容はリセットされる）。

ウィンドウのコンテキストメニューを表示するには、カラムヘッダをクリックします。



	アクション	テーブル	ユーザ	日付	追加	時間
6176	更新	住所録	Designer	2004/12/20	追加	15:20:10
6177	更新	住所録	Designer	2004/12/20	置き換え	15:20:19
6178	削除	住所録	Designer	2004/12/22		15:21:20
6179	追加	住所録	Designer	2004/12/22		15:26:39
6180	追加	住所録	Designer	2004/12/22		15:26:39

次のコマンドを使用できます。

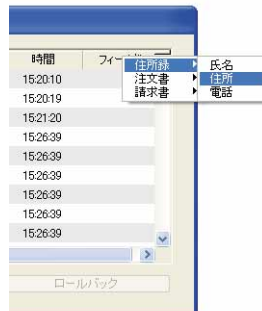
- **追加**：既存のカラムの右側にカラムを追加します。デフォルトでは、カラムは空白であり、「フィールド」という名前が付けられます。ここにデータベースのフィールドを割り当てることができます（次の節を参照）。
- **削除**：ウインドウからカラムを削除します。
- **置き換え**：カレントカラムの内容を置き換えます。標準の情報フィールド（アクション、テーブルなど）のいずれかを選択するか、または「フィールド」を選択してデータベースフィールドの値を表示するカラムを追加することができます。

フィールド値を表示する

ログファイルウインドウ上のカラムにデータベースフィールドを割り当てることができます。これにより、記録された各操作に対するフィールドの現在値が表示されます。この機能は非常に便利であり、例えば、特定のレコードが追加された時点を特定することができます。

これを行うには、コンテキストメニュー（前述）の「追加」または「置き換え」コマンドを用いて、ログファイルウインドウに空の「フィールド」カラムを追加します。

次に、「フィールド」カラムのヘッダをクリックします。すると、データベースのテーブルやフィールドの一覧を含む階層メニューが表示されます。



表示しようとする値を含むフィールドを選択します。すると、追加や修正が行われた各レコードの値が表示されます。

ID	アクション	テーブル	ユーザ	日付	時間	フィールド
6186	追加	住所録	Designer	2004/12/22	15:26:39	梶原和久
6196	追加	住所録	Designer	2004/12/22	15:26:39	梅田周次
6187	追加	住所録	Designer	2004/12/22	15:26:39	梅澤純
6188	追加	住所録	Designer	2004/12/22	15:26:39	梅田唐次
6189	追加	住所録	Designer	2004/12/22	15:26:39	梅田和久
6190	追加	住所録	Designer	2004/12/22	15:26:39	森ひろみ
6191	追加	住所録	Designer	2004/12/22	15:26:39	山本隆一
6192	追加	住所録	Designer	2004/12/22	15:26:39	森下司
6193	追加	住所録	Designer	2004/12/22	15:26:39	平野典二

ペインの表示

ログウインドウに表示するペイン（枠）を作成することができます。これを行うには、ウインドウの上部にあるスプリッタをクリックし、それを下方向へスライドさせます。

ペインの表示

ID	アクション	テーブル	ユーザ	日付	時間
6176	更新	住所録	Designer	2004/12/22	15:20:10
6177	更新	住所録	Designer	2004/12/22	15:20:19
6178	削除	住所録	Designer	2004/12/22	15:21:20
6179	追加	住所録	Designer	2004/12/22	15:26:39
6180	追加	住所録	Designer	2004/12/22	15:26:39
12348	追加	住所録	Designer	2004/12/22	15:26:40
12349	削除	住所録	Designer	2004/12/22	15:27:11
12350	削除	住所録	Designer	2004/12/22	15:27:15
12351	削除	住所録	Designer	2004/12/22	15:28:43

複数のペインを作成するには、この操作を必要なだけ繰り返します。スプリッタをドラッグすることにより、既存のペインの下側にペインを作成することができます。

ペインを削除するには、下側のスプリッタを上側へドラッグするか、スプリッタをダブルクリックします。

データの復元

4th Dimension に組み込まれたバックアップモジュールを使用すると、障害の発生時に、その原因に関わらず、データベースのデータ全体を復元することができます。

障害と診断

発生する可能性のある障害は、主に2種類です。

■ 使用中のデータベースの予期しない中断

この障害が発生する原因としては、停電、システム構成要素の不具合などがあります。この場合、障害が発生した時点のデータキャッシュの状態に応じ、データ復元のためには異なる操作が必要となります。

■ キャッシュが空の場合：データベースは通常通りオープンします。データベースで行われた変更はすべて記録されています。この場合、処理は特に必要ありません。

■ キャッシュに操作が含まれている場合：データベースは損傷していませんが、カレントログファイルの統合が必要です。

■ キャッシュへの書き込み中であった場合：データファイルはおそらく損傷しています。前回のバックアップを復元し、カレントログファイルを統合しなくてはなりません。

■ データベースファイルの損失

この障害が発生する原因としては、データベースが格納されているディスク上の不良セクタ、ウイルス、操作エラーなどがあります。

前回のバックアップを復元し、カレントログファイルを統合しなくてはなりません。

障害の後にデータベースが損傷したかどうかを見分けるには、4th Dimensionを使用してデータベースを再起動します。プログラムはセルフチェックを実行し、実行が必要となる復旧処理を詳しく記述します。自動モードの場合、ユーザ側が何も行わなくても、これらの処理は直接実行されます（後述の「自動復元」の節を参照）。

定期バックアップ計画が設定されている場合、4th Dimensionの復元用ツールにより障害が発生する前のデータベースの正確な状態を知ることができます（ほとんどの場合）。主に2つのツールを利用可能です。

■ バックアップの復元

■ ログファイルの統合

これら2つの機能を組み合わせたり、自動化することができます。

自動復元

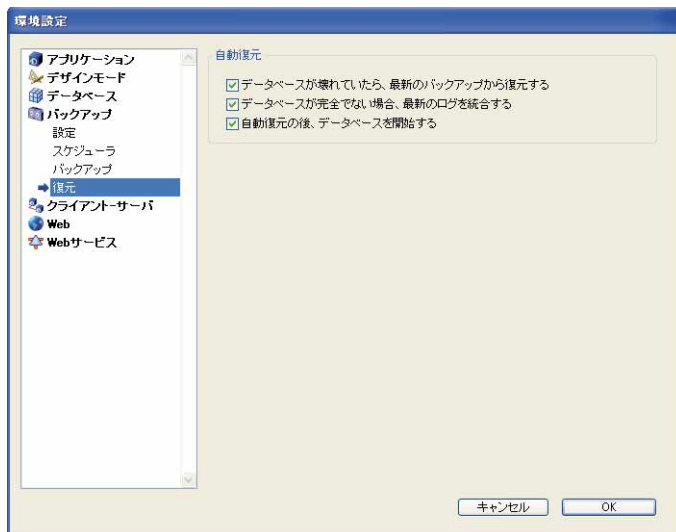
デフォルトとして、4th Dimensionは障害発生後にデータベースの復元手順を自動的に開始します。次の3種類の自動処理が使用されます。

■ 自動復元

■ 自動ログファイル統合

■ 復元後のデータベースの自動再起動

これらのメカニズムは、アプリケーションの「環境設定」の「復元」ページ（「バックアップ」テーマ）にあるオプションを使い、無効にすることができます。



データベースが壊れていたら最新バックアップから復元する

このオプションを選択した場合、データベースの起動中に異常を検出すると、プログラムはそのデータベースの有効な最新バックアップからデータファイルを自動的に復元開始します。ユーザ側の操作は必要ありません。処理はバックアップログに記録されます（後述の「バックアップジャーナル」の節を参照）。

注：

- ・バックアップの復元処理は、1つ以上のバックアップファイルに格納された各ファイルを取り出し、それらを作成したアプリケーションで読み込める状態に戻します。
- ・自動復元の場合、データファイルだけが復元されます。同封ファイルやストラクチャファイルを取り出したい場合は、手動で復元作業を実行しなくてはなりません。

自動復元手順は次の通りです。

- まず始めに、4th Dimension は損傷したデータファイルの名前を変更します。
- 4th Dimension は最新のバックアップからデータファイルを取り出して、それを以前のデータファイルの場所に保存します。
- 「データベースが完全でない場合、最新のログを統合する」オプションが選択されている場合、4th Dimension は必要に応じてログファイルを統合します（後述の「データベースが完全でない場合、最新のログを統合する」の節を参照）。

- 「自動復元の後、データベースを開始する」オプションが選択されている場合、4th Dimension は復元後のデータベースを再起動します（後述の「自動復元の後、データベースを開始する」の節を参照）。

データベースが完全でない場合、最新のログを統合する

このオプションを選択すると、プログラムは復元後のデータベースを開く際に、ログファイルを自動的に統合します。ユーザ側の操作は必要ありません。

- データベースのオープン時に、4th Dimension はログファイルに保存された操作がデータに存在しないことを発見すると、カレントログファイルを自動的に統合します。例えば、データキャッシュに操作が存在しており、それがまだログに書き込まれていない時点で停電が起きると、この状況が発生します。
- データベースを復元する際、同一フォルダ内にバックアップファイルと同じ番号を持つカレントログファイルやバックアップログファイルが保存されている場合、4th Dimension はその内容を検証します。データファイルに存在しない処理が含まれていれば、プログラムはそれを自動的に統合します。

ダイアログボックスは一切表示されず、処理は完全に自動化されています。この機能の目的は、できる限り利用を簡単にすることです。処理はバックアップログに記録されません（後述の「バックアップジャーナル」の節を参照）。

自動復元の後、データベースを開始する

このオプションを選択すると、手順終了後に4th Dimension は復元されたばかりのデータベースを自動的にオープンします。

このオプションを使用した場合に注意すべき点は、データベースの動作に必要なすべての要素が復元後に利用可能であることが重要だということです。例えば、データベースにおいて、そのストラクチャファイルと同じ階層にある特定ファイルが必要となる場合、それらのファイルはデータベースと一緒に復元されるようにバックアップ内に格納しておかなければなりません。データベースでプラグインを使用する場合、それも4th Dimension アプリケーションと同じ階層にインストールしておく方がよいでしょう。

これら3つの復元オプションを組み合わせると（バックアップオプションと連携して）、障害発生後に回復を行う半自動式または自動式のシステムを設定することができます。

- **半自動式**：管理者がデータベースを再起動するだけで、4th Dimension アプリケーションは必要な復元処理を自動的に実行し、信頼できるバージョンのアプリケーション（更新済）を再起動します。

■ **自動式**（4D Serverのみ）：システム起動時に開始するサービスとしてデータベースを保存している場合（前述の「Mac OS Xサービスとして4D Serverを登録する」の節を参照）、そのデータベースはマシンの再起動時に自動的に再開されます。4th Dimensionアプリケーションは必要な復元処理を自動的に行います。

このタイプのシステムは、特に4D Serverデータベースによく適応しています。自動実行される各処理は、データベースのバックアップログファイルに記録される点に注意してください（前述の「データベースが完全でない場合、最新のログを統合する」の節を参照）。

手動でバックアップを復元する

バックアップモジュールによって生成されたアーカイブの内容を手動で復元することができます。例えば、アーカイブの全内容を複製したり（ストラクチャファイルや、同封の付属ファイル）、アーカイブ間での検索を実行するために、手動による復元が必要となる場合があります。

手動による復元作業は、カレントログファイルの統合と一緒に実行することもできます。

注：あらゆる障害が発生した場合に、データベースのデータが自動的に復元されるように4th Dimensionを設定することができます。この件に関する詳細は、前述の「自動復元」の節を参照してください。

4Dアプリケーション自体（4th Dimension、4D Server、4D Engineを組み込んだアプリケーションなど）が、「ようこそ」ダイアログボックスの「復元」ページを用いてバックアップの復元を実行します。

▼ データベースを手動で復元するには、次の手順に従ってください。

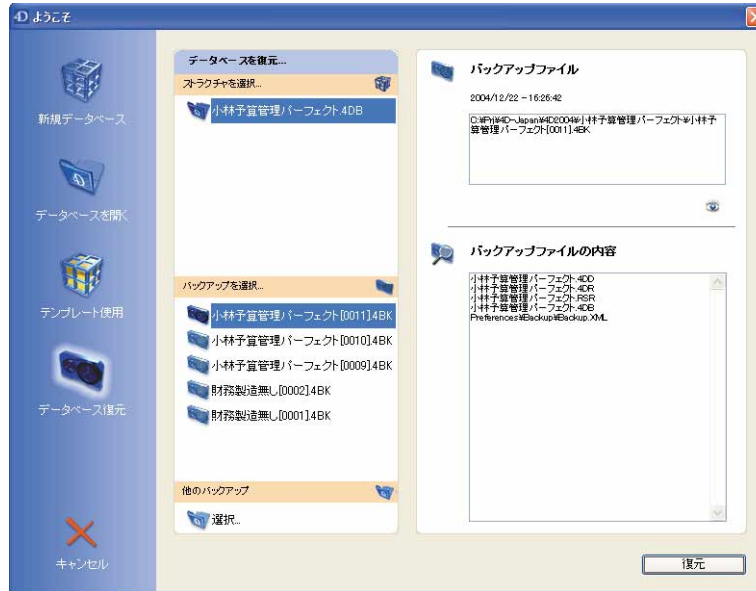
1 「ようこそ」ダイアログボックスにおいて、「復元」タブをクリックする（4th Dimensionおよび4D Serverのみ）。

または、4th DimensionメソッドでRESTOREコマンドを実行する（すべての4Dアプリケーション）。

または、復元するバックアップファイルをアプリケーションアイコン上にドラッグ&ドロップする（すべての4Dアプリケーション）。

後者の場合、標準の「ファイル保存」ダイアログボックスが表示され、ファイルが復元される場所を指定することができます。直接、手順3に進んでください。

2 「最新のアーカイブ」エリアにおいて、データベースを選択した後、復元するバックアップを選び「復元...」をクリックする。



ダイアログボックスのオプションエリアには、選択したアーカイブの内容とアクセスパスが表示されます。「表示する」をクリックすると、ディスク上のバックアップファイルの場所を調べることができます。バックアップがセグメント化されている場合、1番目のセグメントのアクセスパスが表示されます。

または別のアーカイブを復元するには、「他を復元...」をクリックし、標準の「ドキュメントを開く」ダイアログボックスにおいて、復元するバックアップファイルを指定する。

ダイアログボックスが表示され、ファイルが復元される場所を指定することができます。



デフォルトとして、4th Dimensionはアーカイブと同じ階層にある「アーカイブ名- Restore」という名前のフォルダ内に各ファイルを復元します。別の場所を指定するには、「...」ボタンをクリックします。

- 3 必要があれば、復元後のファイルの場所を変更して、「復元」ボタンをクリックします。4th Dimensionはすべてのバックアップファイルを取り出して、指定した場所に配置します。

同じフォルダ内にバックアップファイルと同じ番号のカレントログファイルやバックアップログファイルが保存されている場合、4th Dimensionはその内容を検証します。データファイルに存在しない操作が含まれていれば、プログラムはこれらの操作を統合するかどうかを尋ねてきます。

注：最新ログを統合するオプションを選択している場合、統合は自動的に行われます（前述の「データベースが完全でない場合、最新のログを統合する」の節を参照）。

4 (任意) 「OK」 をクリックして、復元済のデータベースにログファイルを統合する。

復元と統合処理が正常に終了すると、4th Dimensionはダイアログボックスを表示して、データベースをオープンするかどうかを尋ねてきます。

5 「OK」 をクリックして、復元済のデータベースを起動する。

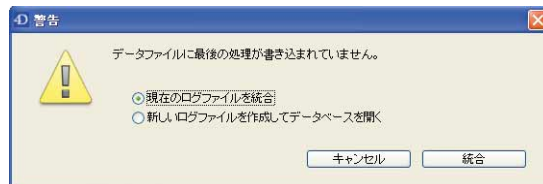
または、「キャンセル」 をクリックして、4th Dimensionアプリケーションを終了する。

注：自動再起動オプションを選択している場合、データベースは自動的に起動します（前述の「自動復元の後、データベースを開始する」の節を参照）。

バックアップ開始時にオリジナルファイルがディスク上のいずれの場所に置かれていようと、4th Dimensionは復元作業中に、すべてのバックアップファイルを目的のフォルダ内に配置します。したがって、使用するファイルを簡単に見つけることができます。

手動でログを統合する

ログファイル自動統合のオプションを選択していない場合（前述の「データベースが完全でない場合、最新のログを統合する」の節を参照）、4th Dimensionはデータベースで実行された操作よりも多くの操作がログファイル内に含まれていることを発見すると、データベースのオープン時に警告ダイアログボックスを表示します。



このメカニズムを有効にするには、4th Dimensionが現在の場所からログファイルへアクセスできなくてはなりません。

カレントログファイルの統合を行うかどうかを選択することができます。カレントログファイルを統合しない場合、データのエラーを再現できなくなります。

処理の取り消し

4th Dimensionには、ログファイル上に実行された操作を取り消せる機能が導入されました。この機能は、複数レベルのキャンセルと同じです。誤ってデータベースからレコードを削除した場合には、この機能が特に役立ちます。

この機能を有効にするためには、データベースでログファイルを使用しなくてはなりません。

▼ データベースを以前の状態に復元するには、次の手順に従ってください。

- 1 4th Dimensionまたは4D Clientの「ファイル」メニューから、「ログファイルをチェック...」コマンドを選択する（サーバ側では、この機能は使用できない）。

ログファイルを操作するダイアログボックスが表示されます。このウィンドウについては、前述の「ログファイルの解析」の節で説明しています。

- 2 それ以降の操作をキャンセルする行を選択する。

選択した行までの操作が残されます。例えば、削除処理をキャンセルしたい場合、削除処理の直前にある操作を選択します。削除処理ならびにそれ以降の操作がすべてキャンセルされます。



- 3 「ロールバック」をクリックする。

4th Dimensionは処理の確定を要求します。

- 4 「OK」をクリックする。

データは、選択した動作の時点の状態まで正確に復元されます。

使用原則

「ロールバック」ボタンをクリックすると、4th Dimensionは現在のデータベースを終了して、データベースデータの最新バックアップを復元します。この後、復元されたデータベースが開かれ、4th Dimensionは選択した処理の時点までログファイルの操作を統合します。データベースがまだ保存されていない場合、4th Dimensionは空のデータファイルを用いて開始します。

バックアップジャーナル

データベースのバックアップの追跡や検証を容易にするため、バックアップモジュールは実行された各処理の概要を特別なファイルに書き込みます。このファイルは、いわゆる活動記録のようなものです。内蔵マニュアルのように、すべてのデータベース操作（バックアップ、復元、ログファイルの統合）は、その処理が定期的または手動のいずれで行われていても、このファイルに記録されます。これらの処理が行われた日付と時間もまた、このジャーナルに記述されます。

次のバックアップの例では、ジャーナルには以下の情報が記述されます。

各アーカイブに関して

 アクセスパス（アーカイブ、または作成、削除されたログへの）

 サイズ

 アーカイブの内容（データファイル、ストラクチャファイル、ユーザストラクチャファイル、インデックスファイル、ログファイル）

アーカイブの各項目に関して

 ファイルタイプ

 オリジナルのファイルサイズ

 日付/時刻（開始時刻、予定時刻、終了時刻）

バックアップジャーナルには“Journal.txt”という名前が付けられ、データベースのデータファイルと同じ階層に配置されます。

バックアップジャーナルは、任意のテキストエディタで開くことができます。解析を容易にするため、ジャーナルの情報はタブで区切られ、情報の各行はキャリッジリターンで終わります。

XML パラメータファイル

4th Dimension のバックアップおよび復元のプレファレンスはすべて、独立した XML ファイルとして提供されます。4th Dimension はこのファイル内のデータを使用して、各バックアップの開始時に、「環境設定」ダイアログボックスのオプションを表示します。

4th Dimension の XML コマンド（後述の「XML」の節を参照）を使用するか、あるいは XML エディタを利用して、バックアップ設定の取得や変更を行うことができます。

これにより、特にコンパイルして 4D Engine を組み込んだアプリケーションを使用している場合に、開発者はプログラムからバックアップ設定を管理することができます。

Web ブラウザ上での
バックアップ設定
XML ファイルの表示

このファイルには、2組のプレファレンスが納められています。

- 一般設定：一般ならびに上級バックアップ設定
- データファイル特定の設定：前回のバックアップに関連する設定（前回のバックアップの日付と時刻、次回バックアップの日付と時刻、前回バックアップファイルの場所、前回バックアップの番号）

XML のバックアップと復元設定ファイルの名前は Backup.XML です。このファイルは次の場所に自動的に作成されます。

DatabaseFolder/Preferences/Backup/Backup.xml

「DatabaseFolder」はデータベースのストラクチャファイルを含むフォルダです。サブフォルダである「Preferences/Backup/」は、必要に応じて自動的に作成されます。

4D Backup について

この節は、4th Dimension のバックアップおよび復元用のプラグインである 4D Backup ユーザを対象としています。この節では 4th Dimension 2004 の新しいバックアップモジュールの機能と、4D Backup の機能を比較します。

注：4D Backup は、バージョン 2004 より前の 4th Dimension との互換性がありません。

互換性

4th Dimension 2004 のバックアップモジュールで使用されているアルゴリズムやメカニズムはまったく新しいものです。この結果、4D Backup のバックアッププロジェクトは、4th Dimension 2004 のバックアップモジュールとの互換性がありません。

また、4D Backup で作成されたアーカイブをバックアップモジュールで復元することはできません（その逆も同様）。

一方で、ログファイルの構造は変更されていません。以前のバージョンの 4th Dimension で作成されたログファイルは、現在のバージョンで使用することができます。

サポートされない 4D Backup の機能

この節では、新しいバックアップモジュールでは利用できない 4D Backup の機能を一覧します。一部の機能は、このモジュールの将来のバージョンに追加される予定です。

■ 各種サポートに関する特定の処理（ディスクット、リムーバブルボリューム、DAT など）。

新しいバックアップモジュールは、オペレーティングシステムにより表示される各サポートを同じ方法で管理します。

■ ディスクスペースが不十分な場合の自動分割。

ユーザの責任において、ディスクスペースがアーカイブの保存に十分であることを確認してください。

■ マルチボリュームバックアップ

新しいバックアップモジュールでは、複数ボリュームにわたりコピーができるように、アーカイブを分割することができます。

■ 論理ミラー

この機能は、新しいバックアップモジュールの最初のバージョンには実装されていません。

新しいバックアップモジュールのメリット

この節では、4D Backup よりも優位性がある新しいバックアップモジュールの主機能について紹介します。

- バックアップ管理の簡略化：デフォルトのバックアップ計画、4th Dimension への統合、同期化したインタフェース
- 異なるプラットフォームや言語間において簡略化され、統一されたバックアップファイルの命名方式
- 4th Dimension、4D Server、4D Engine を組み込んだアプリケーションにおける定期的バックアップと各種オプション
- 4th Dimension、4D Server、4D Engine を組み込んだアプリケーションにおける自動バックアップと復元機能
- データベースのデータに実行された操作に対するマルチレベルキャンセル機能
- バックアップの圧縮と保護（インタレース、リダンダンシー）
- すべてのバックアップや復元のオプションに関し、XML でプログラム化されたアクセス

今回の新リリースでは、4th Dimension 2004 の Web サーバ、4D Server および 4D Client においてさらに多くの新機能が提供されています。

- 持続的接続 (keep alive 接続) の管理
- 暗号化接続 (HTTPS) に使用される TCP ポートの変更
- 未知の URL サポートに関する変更
- Mac OS X 上での Web サーバ起動に対する新しいソリューション
- Web サーバ検証のための新しいメニューコマンド
- 3つの新規コマンド：**PROCESS HTML TAGS**、**GET HTTP BODY**、**SEND HTTP RAW DATA**。これらの新規コマンドについては、後述の「Web サーバ」の節で説明しています。
- CGI の取り扱いに関する変更。特に CGI の処理を目的とする新規コマンドの追加。この新しい機能については、後述の「CGI サポート」の節で説明しています。

注：4D Web サーバは HTTP/1.1 を完全にサポートするようになりました。

持続的接続 (keep alive 接続)

HTTP/1.1 を使用する上での主要な利点の一つは、4th Dimension Web サーバで持続的接続を利用できるようになった点です。

4th Dimension 2004 の Web サーバ、4D Server 2004、4D Client 2004 は、HTTP/1.1 プロトコルを完全にサポートします。

持続的接続を使用する理由

HTTP/1.0において、TCP接続はサーバにより各リクエストごとに開かれては閉じられます。例えば、あるHTMLページ内に10個のピクチャが含まれている場合、Webブラウザでそのページを開くことにより、サーバ上ではTCP接続の開閉が11回行われます。

しかし、TCP接続の開閉作業により、一定量のCPU時間、処理能力、Webサーバメモリが消費されます。

「keep alive接続」オプションを使用すると、1回のTCP接続を維持したままWebブラウザとサーバの間で一連のやり取りを行い、システム資源の節約や転送の最適化を図ることができます。

keep alive 接続を使用する

この新しいオプションは、4th Dimensionの「環境設定」の「Web」テーマにある「オプション」ページにおいて利用することができます。このオプションにより、持続的なTCP接続を有効または無効に設定することができます。



デフォルトとして、このオプションは有効に設定されています。たいいてい場合は、このオプションを選択しておくことをお勧めします。Webブラウザが持続的接続をサポートしていない場合、4D Webサーバは自動的にHTTP/1.0へ切り替わります。

4th Dimension Web サーバのkeep-alive 機能は、コンテキストモードおよび非コンテキストモードにおいて、あらゆる TCP/IP 接続（HTTP、HTTPS）に影響します。ただし、持続的接続はすべての 4D Web プロセスに対して使用されるとは限らない点に注意してください。場合によって、他の最適化された内部機能が呼び出されることがあります。持続的接続は、主としてスタティックページで役立ちます。

「接続毎のリクエスト数」ならびに「タイムアウト」オプションを使用して、このメカニズムの動作を設定することができます。

- **接続毎のリクエスト数**：持続的接続においてやり取りを行えるリクエストと応答の最大数を設定することができます。接続ごとのリクエスト数を制限することにより、多数のリクエスト受信が原因となるサーバの混雑（フラッディング：ハッカーが利用するテクニック）を避けることができます。

4D Web サーバをホスティングするマシン資源（リソース）に応じて、デフォルト値（100）を増やしたり、または減らすことができます。

- **タイムアウト**：この値により、最長待機時間（秒単位）を設定します。この間、Web サーバは Web ブラウザからのリクエストを受け取らない場合でも、開かれた TCP 接続を持続します。この時間に達すると、サーバはその接続を閉鎖します。

接続閉鎖後に Web ブラウザがリクエストを送信した場合、サーバは接続が閉じられたことを知らせます。次に、Web ブラウザは接続がタイムアウトであることを示すメッセージを表示して、ユーザに対し Web ブラウザ上の「更新」ボタンを用いて再接続するよう求めます。すると、新しい接続が開始されます。

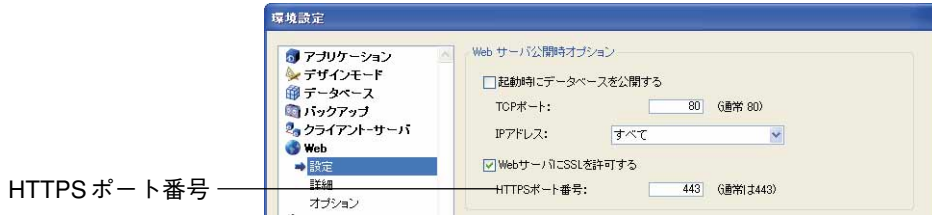
HTTPS のポート番号を変更する

SSL（HTTPS プロトコル）を用いた暗号化 HTTP 接続に対して Web サーバが使用する TCP/IP ポート番号を変更することができます。デフォルトとして、HTTPS ポート番号は 443（標準値）に設定されています。

次の 2 つの理由から、このポート番号の変更を検討する場合があります。

- **セキュリティ上の理由**：ハッカーによる Web サーバへの攻撃は、おおむね標準の TCP ポート（80 と 443）に集中しています。
- **Mac OS X 上で、“一般”ユーザによる暗号化モードでの Web サーバ起動を許可するため**。Mac OS X において、Web 公開用に予約されている TCP/IP ポート（0 から 1023）を使用するには、特定のアクセス権が必要になります。つまり、ルートユーザだけが、これらのポートを使用してアプリケーションの起動を行えます。一般ユーザが Web サーバを開始できるようにするためのソリューションの一つが、TCP/IP ポート番号の変更です。

HTTPS のポート番号は「環境設定」の「Web」テーマにある「設定」ページで変更することができます。



任意の有効な値を渡すことができます。標準のTCPポート番号について知りたい場合は、4D Internet Commands ドキュメントの「付録B、TCPポート番号」の節を参照してください。

未知のURLのサポート

以前のバージョンの4th Dimensionでは、特例として未知のURL（既存のページや4D特定のURLのいずれにも該当しないURL）がサポートされていました。つまり、「On Web Authentication」および「On Web Connection」データベースメソッドにおいて、引数 \$1 に返される値が “/” 文字で始まらない場合です。

例えば、“http://123.4.567.89/Customers/Add” というURLに関して、「Customers」フォルダ内に “Add” というページが存在しない場合、引数 \$1 に “Customers/Add” という値を代入してデータベースメソッドが呼び出されます。

この特殊なケースは問題を引き起こす可能性があるため、4th Dimension 2004ではこれが削除されています。したがって、引数 \$1 に送られるすべてのURLは “/” 文字で始まります。

互換性に関する注意：この特例に基づいた独自のアルゴリズムを採用している開発者がいるかもしれません。このため、「環境設定」の「設定」ページの新しいオプションを使用して、Webサーバが以前の動作を維持するように設定することができます。詳細については、前述の「新しい互換性オプション」の節を参照してください。

Mac OS X 上で Web サーバを開始する

Mac OS Xにおいて、Web公開用に予約されたTCP/IPポート（ポート0から1023）を使用するには、特定のアクセス権が必要になります。つまり、そのマシンのルートユーザだけが、これらのポートを使用してアプリケーションを起動することができます。しかし、“ルート”セッションの使用には、セッション中に生成されるドキュメントのセキュリティやアクセス権など、いくつかの問題があります。

以前のバージョンの4Dではこれ以外のソリューションも利用できましたが（公開用TCPポートの変更、リクエストを受信するポートの転送）、いずれの場合も状況によっては不十分であることがわかりました。

そこで、4th Dimension 2004では、新しいソリューションである一時的ルートセッションが提供されています。この新しいソリューションの機能は次の通りです。まず初めに、4th Dimension Webサーバは、この目的のために開かれた“ルート”セッションにおいて開始されますが、このセッションは開始直後にクローズされます。この新しい機能は、4th Dimension、4D Server、4D Client、および4D Engine実行形式アプリケーションで使用することができます。この操作は、起動時に公開されるデータベースでのみ機能するという点に留意してください。

運用手順を次に説明します。

- 1 4th Dimension アプリケーションが従来のユーザセッションを用いて実行される。
- 2 Webサーバがポート80上で開始されると、警告ダイアログボックスが現れ、その操作が実行不可能であることをユーザに通知する。

このダイアログボックスを使用して、ユーザはWebサーバを開始可能にするためにアクセス権を変更することができます。アクセス権を変更するには、そのマシンの管理者名とパスワードを入力しなくてはなりません。

- 3 ユーザはマシンの管理者名とパスワードを入力する。

4th Dimensionはこの情報を使用して、アプリケーションのアクセス権を変更し、ユーザセッションを“ルート”として設定することができます。

- 4 ユーザはアプリケーションを終了して再起動するよう要求される。

アプリケーションの起動時にWebサーバを開始する必要がある点に留意してください。このため、アプリケーションの「環境設定」において、「起動時にデータベースを公開する」オプションを選択しておかなければなりません。

- 5 起動時に、4th Dimension アプリケーションはルートセッションで開始し、Webサーバはポート80を使って自動的に開始される。

今度は、データベースの公開が正常に行われます。

注：「WebサーバにSSLを許可する」オプションが選択されている場合、SSLポート（デフォルトでは443）もオープンされます。

6 しばらくすると、ルートセッションは自動的に閉じられ、カレントユーザセッションに置き換わります。

この処理はユーザには見えません。Webサーバは公開されたままであり、ユーザセッションは通常通り継続されます。

注：CLIを使用すると、カレントユーザセッション以外のユーザセッションで“ルート”セッションを置き換えることができます（詳細については、前述の「Mac OS Xにおけるコマンドラインインタフェース（CLI）」の節を参照してください。

重要：環境が整った後は、セッション中にWebサーバの中止、開始、再開を任意に行うことができません。また、アプリケーションを再起動しない限り、標準ユーザセッションから“ルート”セッションへ移ることもできません（上位のアクセス権が必要）。

このメカニズムは、アプリケーションの起動時にのみ有効です。

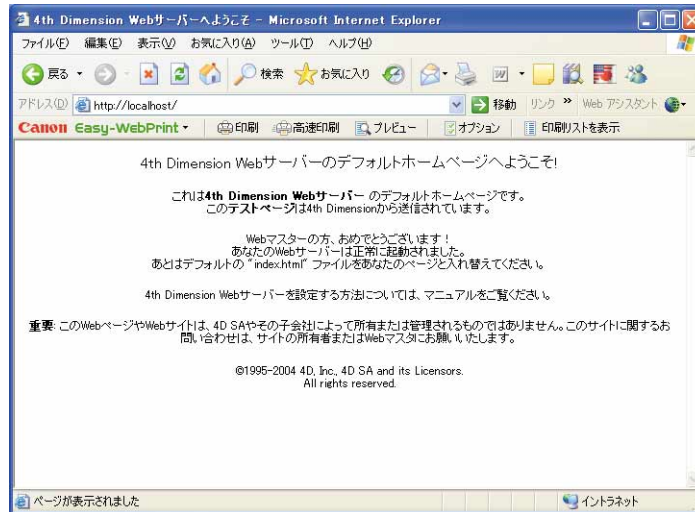
Webサーバの検証

新しい4th Dimensionのメニューコマンドを使用して、内蔵のWebサーバが正常に機能しているかどうかを確認することができます。

このコマンドは、Webサーバが開始している時に、4th Dimension 2004の新しい「実行」メニューから利用することができます。

実行	
コンパイル済み	Ctrl+Shift+K
メソッド ランタイムエクスプローラ	Ctrl+E
Webサーバ停止	
Webサーバリスト	

このコマンドを選択すると、使用しているデフォルトWebブラウザ上に、4Dアプリケーションにより公開されたWebサイトのホームページが表示されます。



このコマンドを使用し、Webサーバやホームページの表示などが正常に機能しているかどうかを検証することができます。

このページは、URL Localhostを用いて呼び出されます。これはWebブラウザが実行されるマシンのIPアドレスを指定するための標準ショートカットです。コマンドは、アプリケーションの「環境設定」で指定されたTCP公開ポート番号を考慮します。

4th Dimension 2004には数多くの新規コマンドや定数、データベースメソッドが追加され、ランゲージ機能がより一層充実しています。さらに、今回の新しいリリースにおいて、既存のコマンドもいくつか機能拡張されています。

- 7つの新しいテーマとともにその新規コマンドが追加され、4th Dimensionの新機能を管理することができます。
- 「フォーミュラ」テーマの新しいコマンドを使用し、「フォーム」エディタをプログラムから管理することができます。
- 「ユーザフォーム」テーマのコマンドを使用し、ユーザが修正できるフォームを管理することができます。
- 「フォーム」テーマのコマンドを使用し、「デザイン」モードならびに従来の「フォームページ」テーマ（このテーマは4th Dimension 2004より削除）で作成されたフォームを管理することができます。
- 新しい「リストボックス」オブジェクトを管理するコマンドは、「リストボックス」テーマに納められています。
- 「バックアップ」テーマには、4th Dimensionの新しいバックアップモジュールに関連するコマンドが納められています。
- 「外部データソース」テーマにはハイレベルODBCコマンドが納められ、これらのコマンドを使用して、4th Dimensionアプリケーションを他のODBCアプリケーションへ接続することができます。
- 「ツール」テーマには、**ENCODE**コマンドや**BUILD APPLICATION**コマンドなどの各種ユーティリティコマンドが納められています。
- 「階層リスト」テーマのコマンドが変更され、選択項目の処理機能が向上しています。
- 4th Dimension 2004では、アクセス権の管理が変更されました。この結果、「ユーザ&グループ」テーマに新しくユーザ管理用コマンドが追加され、既存のコマンドも変更されています。

- 「ウインドウ」テーマが拡張され、開発者は独自のウインドウサイズを設定できるようになりました。
- 「リレート」テーマが拡張されて新規コマンドが追加され、リレートの制御性が向上しています。
- ロールオーバーや3D“ベベル”ボタン、リストボックス、レコードリスト、階層リストを管理するため、「フォームイベント」テーマに新しいフォームイベントが追加されました。
- 「ユーザインタフェース」テーマに、**GET TABLE TITLES**、**GET FIELD TITLES**、**SCROLL LINES** コマンドが新たに追加されました。また、**Pop up menu** 関数が変更されています。
- 「入力制御」テーマには、**EDIT ITEM** コマンドが新しく追加されています。
- 「印刷」テーマの各コマンドは、サーバ側でも利用できるようになりました。
- 「メッセージ」テーマには、**DISPLAY NOTIFICATION** コマンドが新たに追加されています。
- DIALOG** コマンド（「データ入力」テーマ）のデフォルト動作が変更されています。
- 「システムドキュメント」テーマに新しく **Select document** 関数が追加されています。
- 「オブジェクトプロパティ」テーマには、1つの新規コマンドが追加され、既存のコマンドもいくつか変更されています。
- 「4D環境」テーマには、新しい**OPEN 4D PREFERENCES** コマンドが追加されています。このコマンドを使用して、「環境設定」ダイアログボックスの任意のページを開くことができます。
- 「システム環境」テーマには、**SET ENVIRONMENT VARIABLE** と **LAUNCH EXTERNAL PROCESS** という2つの新規コマンドが追加されています。このコマンドを使用すると、4th Dimension とその開発環境の間で情報のやり取りを行うことができます。
- 「(カレント) セレクション」テーマには、**GET HIGHLIGHTED RECORDS** コマンドと **Displayed line number** 関数が新しく追加されています。**HIGHLIGHT RECORDS**、**MODIFY SELECTION**、**DISPLAY SELECTION** コマンドは変更されています。
- 「Webサーバ」テーマには、**PROCESS HTML TAGS**、**GET HTTP BODY**、**SEND HTTP RAW DATA** という3つの新規コマンドが追加されています。**GET WEB FORM VARIABLES** コマンドは変更されています。

- 「配列」テーマに、**Count in array**関数と**APPEND TO ARRAY GET**コマンドが新たに追加されました。
- 「通信コマンド」テーマに、**GET SERIAL PORT MAPPING**コマンドが新しく追加されました。
- 「XML」テーマが大幅に機能強化され、**SAX**コマンドならびに**DOM**コマンドが新たに組み込まれました。これらのコマンドはXMLツリーの作成を行います。
- 「Webサービス (クライアント)」テーマに、新しい**SET WEB SERVICE OPTION**コマンドが追加されました。

フォーミュラ

4th Dimension 2004では「フォーミュラ」エディタが変更され、開発者側の操作性が一層向上すると同時に、カスタマイズしたフォーミュラを構築しやすくなります。今後は、フォーミュラ上で使用できるコマンドだけが表示されます。これらの変更点については、前述の「フォーミュラエディタ」の節で説明しています。

また、新規コマンドも追加されており、開発者が「フォーミュラ」エディタを直接表示したり (**EDIT FORMULA**)、フォーミュラ作成に適合したメソッドを振り分ける (**SET ALLOWED METHODS**および**GET ALLOWED METHODS**) 際に役立ちます。

SET ALLOWED METHODS

SET ALLOWED METHODS (メソッド配列)

引数	タイプ	説明
メソッド配列	文字列配列	→ メソッド名の配列

SET ALLOWED METHODS コマンドを使用して、カレントプロセスの「フォーミュラ」エディタ上に表示されるメソッドを指定することができます。指定したメソッドはコマンドリストの最後に表示され、フォーミュラ上で利用することができます。デフォルトでは (このコマンドを使用しない場合)、「フォーミュラ」エディタ上にメソッドは表示されません。許可されていないメソッド名がフォーミュラで使用されている場合、シNTAXエラーが生成され、フォーミュラの確定はできません。

引数<メソッド配列>には、「フォーミュラ」エディタに提示するメソッドリストを格納する配列の名前を渡します。この配列は、事前に定義しておかなければなりません。

メソッド名にワイルドカード記号 (@) を使用し、許可されるメソッドグループを1つ以上定義することができます。

注：デフォルトとして許可されていない4D コマンドやプラグインコマンドをユーザが呼び出せるようにしたい場合、これらのコマンドを取り扱う特定のメソッドを使用しなくてはなりません。

▼ 次の例題は、名前が“formula”で始まるすべてのメソッドと、“Total_general”メソッドを「フォーミュラ」エディタで許可します。

```
ARRAY STRING(15;methodsArray;2)
methodsArray{1}:="formula@"
methodsArray{2}:="Total_general"
SET ALLOWED METHODS(methodsArray)
```

参照

GET ALLOWED METHODS

GET ALLOWED METHODS

GET ALLOWED METHODS (メソッド配列)

引数	タイプ	説明
メソッド配列	文字列配列	← メソッド名の配列

GET ALLOWED METHODS コマンドは、フォーミュラの作成に使用できるメソッド名を引数<メソッド配列>に返します。これらのメソッドは、エディタ上のコマンドリストの最後に一覧表示されます。

デフォルトでは、「フォーミュラ」エディタ上でメソッドを使用することはできません。各メソッドは、**SET ALLOWED METHODS** コマンドを使用して明示的に許可しなくてはなりません。このコマンドが実行されていない場合、**GET ALLOWED METHODS** コマンドは空の配列を返します。

GET ALLOWED METHODS コマンドは、**SET ALLOWED METHODS** コマンドに渡されたものとまったく同じ値を返します。必要があれば、コマンドは配列の作成やサイズ調整を行います。また、ワイルドカード記号 (@) を使用してメソッドグループが設定されている場合、“@” 記号を含む文字列が返されます (メソッドグループの名前ではありません)。

▼ この例題では、レポート作成のため一連の特定メソッドを許可します。

```
GET ALLOWED METHODS(methodsArray)
  `クイックレポート用のメソッドを定義する
ARRAY TEXT(reports_methods;1)
reports_methods{1}:="Reports_@"
SET ALLOWED METHODS(reports_methods)
```



```
QR REPORT([従業員];"MyReport")
  ` カレントパラメータを再設定する
SET ALLOWED METHODS(methodsArray)
```

参照

SET ALLOWED METHODS

EDIT FORMULA

EDIT FORMULA (テーブル; フォーミュラ)

引数	タイプ	説明
テーブル	テーブル	→ デフォルトとしてフォーミュラエディタに表示するテーブル
フォーミュラ	文字列変数	→ フォーミュラエディタに表示するフォーミュラを格納する変数または、エディタだけを表示する場合は "" ← ユーザが確定したフォーミュラ

EDIT FORMULA コマンドを使用すると、「フォーミュラ」エディタを開いて次の項目をデフォルトとして表示することができます。

- 左側のリストには、引数<テーブル>に渡したテーブルのフィールドが表示されます。
- フォーミュラエリアには、変数<フォーミュラ>に格納されたフォーミュラが表示されます。<フォーミュラ>に空の文字列を渡した場合、フォーミュラは指定せずに「フォーミュラ」エディタが表示されます。

エディタが表示された後は、<フォーミュラ>をそのままの状態で行実行可能ですが（構文が正しい場合）、その修正やフォーミュラの作成、新しいフォーミュラのロード、保存を行うこともできます。いずれの場合でも、ユーザがダイアログボックスを確定すると、システム変数OKには1が代入され、変数<フォーミュラ>には実際に実行されたフォーミュラが格納されます。ユーザがフォーミュラをキャンセルすると、システム変数OKには0が代入され、変数<フォーミュラ>は変更されません。

<フォーミュラ>でメソッドが呼び出され、そのメソッドが**SET ALLOWED METHODS** コマンドを用いて「フォーミュラ」エディタ上で事前に“許可”されていない場合は、シンタックスエラーが生成され、フォーミュラを確定することができません。

- ▼ 事前に入力されたフォーミュラは指定せず、[従業員]テーブルを使用してフォーミュラエディタを表示します。

```
$myFormula:=""
EDIT FORMULA([従業員];$myFormula)
```

参照

SET ALLOWED METHODS

ユーザフォーム

4th Dimension 2004において、開発者はユーザに対して独自のフォームの作成や変更を実行する機会を与えることができます。その結果、“ユーザフォーム”を他の4th Dimension フォームと同じように使用することができます。

ユーザフォーム入門

ユーザフォームは開発者が「デザイン」モードで作成した4th Dimension 標準のフォーム（“ソース”フォームまたは“ディベロッパ”フォーム）をベースにしており、そのフォームには「ユーザ編集可」プロパティが適用されています。簡易版の「フォーム」エディタ（**EDIT FORM** コマンドで呼び出される）を使用して、ユーザはフォームの外観やグラフィックオブジェクト、非表示項目などを修正することができます。また、開発者は、認可する動作を制御することができます。「ユーザフォーム」エディタに関する詳細については、前述の「ユーザフォームエディタ」の節を参照してください。

ユーザフォームは、次の2通りの方法で使用することができます。

- ユーザは**EDIT FORM** コマンドを使用して“ソース”フォームを修正し、独自のニーズに合わせます。

ユーザフォームはローカル上に保持され、オリジナルフォームの代わりとして自動的に使用されます。

この動作は、例えば、フォームに会社ロゴを追加したり、不要なフィールドを隠すなど、開発者がダイアログボックスのパラメータを現場で設定する必要性に対処するものです。

- “ソース”ファイルは基本テンプレートとして機能し、ユーザは**CREATE USER FORM** コマンドを使用して、いくつでも自由にコピーを作成することができます。**EDIT FORM** コマンドを使用すると、それぞれのコピーの設定を自由に行えます（内容、名前など）。ただし、各ユーザフォームの名前はユニークでなくてはなりません。これらのコピーを利用するには、**INPUT FORM** コマンドと**OUTPUT FORM** コマンドを使用します。これらのコマンドは引数としてユーザフォームを受け入れるようになりました。

この対応方法により、例えば開発者はユーザ用にカスタマイズしたレポートを作成することができます。

ユーザフォームとライブラリオブジェクト

「ユーザフォーム」エディタにおいて、4th Dimension のオブジェクトライブラリのオブジェクトを挿入することができます。オブジェクトライブラリに関する詳細は、前述の「オブジェクトライブラリ」の節を参照してください。

オブジェクトライブラリを「ユーザフォーム」エディタで使用すると、オブジェクトと一緒にそのグラフィックプロパティや自動アクションもペーストすることができます。メソッドが付属するオブジェクトは、ライブラリ上に表示されません。

注意すべき点は、追加するライブラリオブジェクトが名前や変数、タイプの点でユーザフォーム（および、そのオブジェクト）と互換性があるかどうかを確認するのは、開発者の責任であるということです。

ユーザフォームの保存と管理

ユーザフォームのメカニズムは、4th Dimension、4D Server、4D Runtime、または4D Engine を使用したコンパイル済みデータベースおよびインタプリタ版のデータベースの双方で機能します。クライアント/サーバモードでは、ユーザが修正したフォームをすべてのマシンで利用することができます。

4D が自動的にフォームの変更を処理します。ユーザが「ユーザ編集可」としてフォームを設定している場合、「デザイン」モードにおいてそのフォームがロックされます。フォームオブジェクトへアクセスできるようにするには、開発者がアイコンをクリックして、そのフォームのロックを明示的に解除しなければなりません。この操作により、関連するユーザフォームが古くなりますが、その場合は再作成しなくてはなりません。

“ソース”フォームが削除されると、関連するユーザフォームも一緒に削除されます。

ユーザフォームは、拡張子 “.4DA” が付けられた別ファイルに保存され、メインのストラクチャファイル (.4DB/.4DC) と同じ階層に配置されます。このファイルの機能は透過的です。つまり、4th Dimension はユーザフォームが存在すれば、それを使用します (**LIST USER FORMS** コマンドを使用し、有効なユーザフォームをいつでも探すことができます)。ユーザフォームが古ければ、それが削除され、4th Dimension はデフォルトとしてソースフォームを使用します。

クライアント/サーバでは、メインのストラクチャファイルと同じ規則に従って、「.4DA」ファイルが各クライアントマシンへ送信されます。

また、この原理により、開発者がストラクチャファイルの更新を行った場合でも、ユーザフォームの陳腐化を避けることができます。

EDIT FORM

EDIT FORM (テーブル; フォーム{; ユーザフォーム}{; ライブラリ})

引数	タイプ	説明
テーブル	テーブル	→ 変更するフォームの属すテーブル
フォーム	文字列	→ 変更するフォームの名前
ユーザフォーム	文字列	→ 修正するユーザフォームの名前
ライブラリ	文字列	→ 利用可能なオブジェクトライブラリのパス名

EDIT FORM コマンドは、引数<テーブル>、<フォーム>および任意の引数である<ユーザフォーム>を用いて指定されたフォームを「ユーザフォーム」エディタ上に開きます。



注：そのプロセスの最初のウィンドウである場合にのみ、エディタウィンドウが開かれます。言い換えれば、エディタを表示するには常に新規プロセスをオープンする必要がありますということです。

ユーザがフォームへの変更を保存すると、システム変数OKに値“1”が代入されます。

引数<ユーザフォーム>を渡さず、しかも<フォーム>に関連付けられたユーザフォームがもともと存在していない場合、エディタ上にはソースフォームが表示されます。この後、変更後のフォームがユーザストラクチャファイル (.4DA) にコピーされ、<フォーム>の代わりに使用されます。

このコマンドを用いて、ユーザフォームが<フォーム>を元にして既に生成されている場合、エディタ上にユーザフォームが表示されます。ソースフォームから始めたい場合は、まず最初に **DELETE USER FORM** コマンドを使用してユーザフォームを削除しなければなりません。

このコマンドを用いて、ユーザフォームが<フォーム>を元にして既に生成されている場合、エディタ上にユーザフォームが表示されます。ソースフォームから始めたい場合は、まず最初に**DELETE USER FORM**コマンドを使用してユーザフォームを削除しなければなりません。

引数<ユーザフォーム>を使用すると、変更するユーザフォーム (**CREATE USER FORM** コマンドを用いて作成済み) を設定することができます。この場合、そのフォームがエディタ上に表示されます。

<ライブラリ>には、フォームのカスタマイズのためにユーザが使用を許可されるオブジェクトライブラリのフルパス名を渡します。クライアント/サーバモードでは、このライブラリがすべてのクライアントマシンで使用できるように、「Plugins」フォルダと同じ階層にあるデータベースの「Extras」フォルダ内に配置しなくてはなりません。そのライブラリが有効であれば、それがフォームウインドウで開かれます。オブジェクトライブラリに関する詳細は、前述の「オブジェクトライブラリ」の節を参照してください。

操作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。次の場合は、システム変数OKに0が代入され、エラーが生成されます。

- ユーザが「デザイン」モードでフォームの変更を行えない場合や、フォームが存在しない場合。
- フォームが既に開かれ、別プロセスで変更されている場合。
- 適切なアクセス権を持たないため、ユーザがそのフォームへアクセスできない場合。

これらのエラーは**ON ERR CALL**コマンドにより設定されたエラー処理用メソッドを用いて遮ることができます。

▼ 次の例題は、ダイアログのフォームの変更をユーザに許可するプロジェクトメソッドです。

```
EDIT FORM([Dialogs];"Welcome";"Lib_Logos.4il")
```

CREATE USER FORM

CREATE USER FORM (テーブル; フォーム; ユーザフォーム)

引数	タイプ	説明
テーブル	テーブル	→ ソースフォームのテーブル
フォーム	文字列	→ ソースフォーム名
ユーザフォーム	文字列	→ 新規ユーザフォームの名前

処理が正常に実行されると、システム変数OKに1が返されます。次の場合は、システム変数OKに0が代入され、エラーが生成されます。

■ <フォーム>が既にユーザフォームである場合。

■ <ユーザフォーム>に指定した名前が、ソースフォームまたは既存のユーザフォームと同じである場合。

■ 適切なアクセス権を持たないため、ユーザがそのフォームへアクセスできない場合。

このエラーは**ON ERR CALL**コマンドにより設定されたエラー処理用メソッドを用いて遮ることができます。

LIST USER FORMS

LIST USER FORMS (テーブル; フォーム; ユーザフォーム配列)

引数	タイプ	説明
テーブル	テーブル	→ ソースフォームのテーブル
フォーム	文字列	→ ソースフォームの名前
ユーザフォーム配列	文字列配列	← ソースフォームを元にしたユーザフォームの名前

このコマンドは、引数<テーブル>と<フォーム>で指定されたディベロッパフォームを元に作成されたユーザフォームの名前を配列<ユーザフォーム配列>に代入します。

ユーザフォームがEDIT FORMコマンドを使用して直接作成された場合、<ユーザフォーム配列>の項目には空の文字列 ("") だけが代入されます。

指定したディベロッパフォームに対するユーザフォームが存在しない場合、この配列は空になります。

DELETE USER FORM

DELETE USER FORM (テーブル; フォーム; ユーザフォーム)

引数	タイプ	説明
テーブル	テーブル	→ ユーザフォームのテーブル
フォーム	文字列	→ ソースフォーム名
ユーザフォーム	文字列	→ ユーザフォーム名

このコマンドを使用し、引数<テーブル>、<フォーム>、<ユーザフォーム>で指定したユーザフォームを削除することができます。

ユーザフォームがEDIT FORMコマンドを使用して直接作成された場合、<ユーザフォーム>に空の文字列（""）を渡してください。

ユーザフォームの削除が正常に終了すると、システム変数OKに1が代入されます。一方、次の場合はシステム変数OKに0が代入され、エラーが生成されます。

■ ユーザフォームが存在しない場合

■ ユーザが十分なアクセス権を持たない場合

このエラーは**ON ERR CALL**コマンドにより設定されたエラー処理用メソッドを用いて遮ることができます。

ユーザフォームのエラーコード

ユーザフォームを管理するコマンドに関連するエラーコードの一覧を次に示します。

-9750	ソースフォームが編集可能ではありません。
-9751	ユーザはソースフォームへアクセスできません。
-9752	ユーザフォームを作成できません。
-9753	ソースフォームが存在しません。
-9754	このコマンドはダイアログウインドウから使用できません。
-9755	ユーザフォームに名前が設定されていません。
-9756	ユーザストラクチャファイルが存在しません。
-9757	ユーザフォームが存在しません。
-9758	ユーザフォームが既に存在しています。
-9759	オブジェクトライブラリを開けません。

階層リスト

4th Dimension 2004では、プログラムからの階層リストの管理方法が変更されました。特に、複数項目選択に関する取り扱いが改善され、位置または参照のいずれかを選択するようになります。

これらの改善により、次のコマンドが変更されています。

SELECT LIST ITEMS BY POSITION (SELECT LIST ITEM)、SELECT LIST ITEMS BY REFERENCE、SET LIST PROPERTIES、GET LIST PROPERTIES、SET LIST ITEM PROPERTIES、GET LIST ITEM PROPERTIES、SET LIST ITEM、GET LIST ITEM、List item parent、Selected list items(Selected list item)、Count list items

また、「入力制御」テーマに**EDIT ITEM**コマンドが追加された点に注目してください。この新しいコマンドを使用すると、特に階層リストのリスト項目をプログラムにより編集モードへ移行することができます。

最後に、新しいフォームイベントが追加され、ユーザアクションの管理が向上しています（後述の「階層リストに関連するフォームイベント」の節を参照）。

SELECT LIST ITEMS BY POSITION (SELECT LIST ITEM)

SELECT LIST ITEMS BY POSITION(リスト; 項目位置{; 位置配列})

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
項目位置	数値	→ 展開されたリスト内の項目の位置
位置配列	数値配列	→ 展開されたリスト内の位置を納めた配列

注：SELECT LIST ITEMS BY POSITIONは、以前のバージョンの4th Dimensionで使用されていたSELECT LIST ITEMコマンドの新しい名前です。SELECT LIST ITEMS BY REFERENCEコマンドとの関連性を明確にし、4th Dimension 2004において複数選択が利用可能である点を強調するため、このコマンドの名前が変更されました。

SELECT LIST ITEMS BY POSITIONコマンドでは、新たに追加された引数（任意）である<位置配列>を使用することができます。この引数により、<リスト>内で複数のリスト項目を同時に選択することができます。

<位置配列>には必ず配列を渡し、その各要素は選択するリスト項目の位置を示します。

重要：階層リスト内で複数のリスト項目を同時に選択するには（手動、あるいはプログラムから）、リストに対して新しいプロパティである「複数選択可」を有効に設定しておかなくてはなりません。このプロパティの定義は、SET LIST PROPERTIESコマンドを使用しています。

▼ 次の例題は、階層リストの2番目、3番目、5番目の項目を選択します。

```
SET LIST PROPERTIES(hList;0;0;18;0;1)
```

```
  `複数選択を許可するには、  
  `最後の引数に1を渡さなくてはならない
```

```
ARRAY LONGINT($tab;3)
```

```
$tab{1}:=2
```

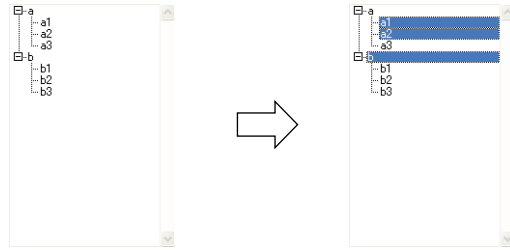
```
$tab{2}:=3
```

```
$tab{3}:=5
```

```
SELECT LIST ITEMS BY POSITION(hList;3;$tab)
```

```
  `3番目の項目がカレントリスト項目として設定される
```

```
REDRAW LIST(hList)
```

SELECT LIST ITEMS BY REFERENCE

SELECT LIST ITEMS BY REFERENCE (リスト; 項目参照{; 参照配列})

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
項目参照	倍長整数	→ 項目参照番号
参照配列	倍長整数配列	→ 項目参照番号の配列

注：SELECT LIST ITEMS BY REFERENCE は、以前のバージョンの 4th Dimension で使用されていた SELECT LIST ITEM BY REFERENCE コマンドの新しい名前です。4th Dimension 2004 において複数選択が利用可能である点を強調するため、このコマンドの名前が変更されました。

SELECT LIST ITEMS BY REFERENCE コマンドでは、新たに追加された引数（任意）である＜参照配列＞を使用することができます。この引数により、＜リスト＞内で複数のリスト項目を同時に選択することができます。

＜参照配列＞には必ず配列を渡し、その各要素は選択するリスト項目の固定参照番号を示します。

重要：階層リスト内で複数のリスト項目を同時に選択するには（手動、あるいはプログラムから）、リストに対して新しいプロパティである「複数選択可」を有効に設定しておかなければなりません。このプロパティの定義は、SET LIST PROPERTIES コマンドを使用して行います（後述）。

引数＜項目参照＞に指定したリスト項目は、そのリストの新しいカレント項目となります。配列に指定した一連のリスト項目にこの項目が属しているかどうかはわかりません。カレントリスト項目とは、具体的に言うと、**EDIT ITEM** コマンドを使用した場合に編集モードに移行される項目のことです。

SET LIST PROPERTIES

SET LIST PROPERTIES (リスト; 表示形式{; アイコン{; 線の高さ{; ダブルクリック{; 複数選択可{; 編集可{}}))

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
表示形式	数値	→ リストの表示形式
アイコン	数値	→ Mac OS ベースのリソースID'icn', または 0=デフォルトプラットフォームの ノードアイコン
線の高さ	数値	→ 線の高さの最小値 (ピクセル単位)
ダブルクリック	倍長整数	→ ダブルクリックで展開/縮小: 0=Yes (デフォルト)、1=No
複数選択可	倍長整数	→ 複数項目の選択: 0=No (デフォルト)、 1=Yes
編集可	倍長整数	→ 0=ユーザによるリスト編集可 (デフォ ルト)、1=ユーザによるリスト編集不可

SET LIST PROPERTIES コマンドには、<複数選択可>および<編集可>という2つの引数 (任意) が追加されています。

■ **複数選択可**: この引数を使用し、<リスト>が複数項目の選択を受け入れるかどうかを指示することができます。

デフォルトでは、以前のバージョンの4th Dimensionと同様に、階層リストの項目を複数同時に選択することはできません。リストに対してこの機能を利用可能にしたい場合は、引数<複数選択可>に値1を渡します。その場合、複数選択機能を使用することができます。

■ **手動の場合**、連続した項目の選択にはShiftキー+クリック、連続していない項目の選択にはCtrlキー+クリック (Windows) またはCommandキー+クリック (Mac OS) を使用します。

■ プログラムを使用する場合、**SELECT LIST ITEMS BY POSITION (SELECT LIST ITEM)** および**SELECT LIST ITEMS BY REFERENCE** コマンドを使用します。

引数<複数選択>に0を渡すか、または省略した場合、デフォルトの動作が適用されます。

■ **編集可**: この引数を使用すると、データ入力中に<リスト>がフィールドや変数に関連付けた選択リストとして表示された時に、ユーザによる<リスト>の編集を可能にするかどうかを指示することができます。<リスト>が編集可である場合、選択リストウインドウに「変更」ボタンが追加され、ユーザは特定のエディタを用いて、値の追加や削除、並び替えを行うことができます。

引数<編集可>に値0を渡すか、または省略すると、そのリストは編集可能になります。1を渡すと、リストを編集することはできません。

参照

GET LIST PROPERTIES

GET LIST PROPERTIES

GET LIST PROPERTIES (リスト; 表示形式{; アイコン{; 線の高さ{; ダブルクリック{; 複数選択可{; 編集可}{})

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
表示形式	数値	← リストの表示形式
アイコン	数値	← Mac OS ベースのリソースID 'cicn'
線の高さ	数値	← 線の高さの最小値 (ピクセル単位)
ダブルクリック	倍長整数	← ダブルクリックでサブリストの展開／縮小：0=Yes、1=No
複数選択可	倍長整数	← 複数項目の選択：0=No、1=Yes
編集可	倍長整数	← 0=ユーザによるリスト編集可 (デフォルト)、1=ユーザによるリスト編集不可

GET LIST PROPERTIES コマンドは、<複数選択>および<編集可>という追加された2つの引数 (任意) を返します。

■ 引数<複数選択可>に0が返された場合、<リスト>においてリスト項目を複数選択することはできません (手動でもプログラムからでも)。1が返された場合は、複数項目を選択することができます。

■ 引数<編集可>に0が返された場合、「ユーザ」モードで選択リストとして表示されると、リストは編集可能になります。1が返された場合、リストを編集することはできません。

これらのパラメータに関する詳細は、前述の **SET LIST PROPERTIES** コマンドを参照してください。

参照

SET LIST PROPERTIES

SET LIST ITEM PROPERTIES

SET LIST ITEM PROPERTIES (リスト; 項目参照 |*; 入力可; スタイル; アイコン {; カラー})

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
項目参照!*	倍長整数!*	→ 項目参照番号、または最後に追加されたリスト項目の場合は0、またはカレントリスト項目の場合は*
入力可	ブール	→ True=入力可、False=入力不可
スタイル	数値	→ 項目のフォントスタイル
アイコン	数値	→ Mac OS ベースのリソース ID 'cicn'または65536+Mac OS ベースのリソース ID 'PICT'、または131072+ピクチャ参照番号
カラー	倍長整数	→ RGB カラーの値、または-1=オリジナルのカラーへリセット

SET LIST ITEM PROPERTIES コマンドには2つの変更点があります。1つは、項目参照として値“*”を受け入れるようになった点、2つ目は、選択したリスト項目の色を変更するための新しいパラメータが追加された点です。

■ 二番目の引数として“*”を渡すことができるようになりました。この場合、コマンドはリストのカレント項目に対して適用されます。複数のリスト項目が手動で選択されている場合、最後に選択された項目がカレントリスト項目となります。選択された項目が存在しない場合、コマンドは何も行いません。

■ このコマンドは、引数<カラー>を受け入れるようになりました。この引数を使用して、テキストや選択したリスト項目の色を変更することができます。

この色はRGB フォーマット、つまり0x00RRGGBB形式の4バイトの倍長整数で定義しなくてはなりません。このフォーマットに関する詳細は、4th Dimension『ランゲージ』マニュアルの**SET RGB COLORS**コマンドの説明を参照してください。

リスト項目をオリジナルの色にリセットするには、引数<カラー>に1を渡します。

▼ 次の例題は、<リスト>のカレント項目のテキストを太字スタイル、明るい赤に設定します。

```
SET LIST ITEM PROPERTIES(list;*;True;Bold;0;0x00FF0000)  
REDRAW LIST(list)
```

参照

GET LIST ITEM PROPERTIES

GET LIST ITEM PROPERTIES**GET LIST ITEM PROPERTIES** (リスト; 項目参照 |*; 入力可 {; スタイル {; アイコン {; カラー }}}

引数	タイプ		説明
リスト	項目参照	→	リスト参照番号
項目参照!*	倍長整数!*	→	項目参照番号、または最後に追加されたリスト項目の場合は0、またはカレントリスト項目の場合は*
入力可	ブール	←	True=入力可、False=入力不可
スタイル	数値	←	項目のフォントスタイル
アイコン	数値	←	Mac OS ベースのリソース ID 'cicn'、または 65536+Mac OS ベースのリソース ID 'PICT'、または 131072+ピクチャ参照番号
カラー	倍長整数	←	RGB カラーの値

GET LIST ITEM PROPERTIES コマンドには2つの変更点があります。1つは、項目参照として値 “*” を受け入れるようになった点、2つ目は選択したリスト項目の色を表わす新しい引数を返す点です。

■ 二番目の引数として “*” を渡すことができるようになりました。この場合、コマンドはリストのカレント項目に対して適用されます。複数のリスト項目が手動で選択されている場合、最後に選択された項目がカレントリスト項目となります。選択された項目が存在しない場合、コマンドは何も行いません。

■ このコマンドは、引数<カラー>を受け入れるようになりました。この引数は、テキストや選択したリスト項目の色を返します。この引数に関する詳細は、**SET LIST ITEM PROPERTIES** コマンドを参照してください。

参照

SET LIST ITEM

SET LIST ITEM

SET LIST ITEM (リスト; 項目参照 |*; 新規項目テキスト; 新規項目参照{; サブリスト; 展開))

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
項目参照!*	倍長整数!*	→ 項目の参照番号、または最後に追加された項目の場合は0、またはそのリストのカレント項目の場合は*
新規項目テキスト	文字列	→ 新規項目のテキスト
新規項目参照	数値	→ 新規項目の参照番号
サブリスト	リスト参照	→ 項目に付属させる新規サブリスト 0=サブリストなし、-1=変更しない
展開	ブール	→ 任意のサブリストを展開するか縮小するかを指示する

SET LIST ITEM コマンドは、項目参照として “*” を受け入れるようになりました。この場合、コマンドはリストのカレント項目に対して適用されます。手動で複数のリスト項目が選択されている場合、最後に選択された項目がカレントリスト項目になります。選択された項目が存在しない場合、コマンドは何も行いません。

参照

GET LIST ITEM

GET LIST ITEM

GET LIST ITEM (リスト; 項目位置 |*; 項目参照; 項目テキスト{; サブリスト{; 展開))

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
項目位置!*	数値!*	→ 展開されたリストにおける項目の位置、またはリストのカレント項目の場合は*
項目参照	倍長整数	← 項目の参照番号
項目テキスト	文字列	← リスト項目のテキスト
サブリスト	リスト参照	← サブリストの参照番号 (存在する場合)
展開	ブール	← サブリストが展開されているか縮小されているかを示す

GET LIST ITEM コマンドは、項目参照として “*” を受け入れるようになりました。この場合、コマンドはリストのカレント項目に対して適用されます。

手動で複数のリスト項目が選択されている場合、最後に選択された項目がカレントリスト項目になります。選択された項目が存在しない場合、コマンドは空の値を返します。

参照

SET LIST ITEM

List item parent

List item parent (リスト; 項目参照 | *) → 倍長整数

引数	タイプ		説明
リスト	リスト参照	→	リスト参照番号
項目参照!*	倍長整数!*	→	項目の参照番号、または リストのカレント項目の場合は*
戻り値	倍長整数	←	親項目の項目参照番号、または 親項目がない場合は0

List item parent関数は、項目参照として“*”を受け入れるようになりました。この場合、この関数はリストのカレント項目に対して適用されます。手動で複数のリスト項目が選択されている場合、最後に選択された項目がカレントリスト項目になります。選択されたリスト項目が存在しない場合、コマンドは何も行いません。

Selected list items (Selected list item)

Selected list items (リスト{; 項目配列}{; *}) → 倍長整数

引数	タイプ		説明
リスト	リスト参照	→	リスト参照番号
項目配列	倍長整数配列	←	“*”を省略した場合、配列には展開されたリストにおける選択項目の位置が納められる “*”を指定した場合、配列には選択項目の参照番号が納められる
*	*	→	省略した場合：項目の位置 指定した場合：項目の参照番号
戻り値	数値	←	“*”を省略した場合：展開されたリスト内で現在選択されているリスト項目の位置 “*”を指定した場合：選択項目の参照番号

注：Selected list items は、以前のバージョンの 4th Dimension で使用されていた Selected list item 関数の新しい名前です。4th Dimension 2004 において複数選択が利用可能である点を強調するため、このコマンドの名前が変更されました。

4th Dimension 2004 では、**Selected list items** 関数の機能が拡張されています。

■ コマンドは、項目の現在位置または項目の固定参照のいずれも処理できるようになりました。

■ また、コマンドは複数の選択項目にも対応します。コマンドは、選択されたすべてのリスト項目の位置あるいは参照番号を格納した配列を返します。

引数<項目配列>には、倍長整数タイプの配列を渡すことができます。コマンドを使用すると、必要に応じて配列の作成やサイズ調整が行われます。コマンドの実行後<項目配列>には次の要素が代入されます。

■ 引数“*”を省略した場合、展開されたリスト内の各選択項目の位置。

■ 引数“*”を渡した場合、各選択項目の固定参照。

選択された項目が存在しない場合、空の配列が返されます。

引数“*”を使用して、項目の位置（この場合、引数“*”は省略）または固定の項目参照（その際、引数“*”を使用）のいずれを処理するかを指示します。

引数“*”に応じて、コマンドは<リスト>のカレント項目の位置、あるいは参照番号を返します。複数の選択項目が存在する場合、カレント項目は、ユーザが最後にクリックした項目（手動による選択の場合）または **SELECT LIST ITEMS BY POSITION (SELECT LIST ITEM)** か **SELECT LIST ITEMS BY REFERENCE** コマンド（プログラムによる選択）で指定された項目のいずれかとなります。

Count list items

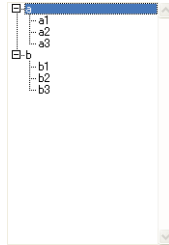
Count list item (リスト{; *}) → 数値

引数	タイプ	説明
リスト	リスト参照	→ リスト参照番号
*	*	→ 省略した場合（デフォルト）：（展開されている）表示リスト項目を返す 指定した場合：すべてのリスト項目を返す
戻り値	数値	← （展開されている）表示リスト項目の数（“*”を省略した場合）、またはリスト項目の合計数（“*”を指定した場合）

Count list items 関数は、リスト参照として “*” を受け入れるようになりました。この引数を渡した場合、リストが展開されているか、縮められているかに関わらず、関数は項目の合計数を<リスト>に代入して返します。

この引数を省略すると、コマンドは展開されている項目の数、つまり表示されているリスト項目の数を返します。

▼ 次のような階層リストの場合



次に示すのは、リストが以下のような状態の場合に **Count list items** から返される値を比較した例題です。



C_LONGINT(\$vLnumber;\$vLlist)

`$vLnumber:=Count list items($vLlist)` ` \$vLnumber は 5

`$vLnumber:=Count list items($vLlist;*)` ` \$vLnumber は 8

ユーザ&グループ

4th Dimension 2004では、アクセス管理およびパスワード管理のシステムが変更されました。この新しいリリースのランゲージにおける主要な変更点は次の通りです。

■ プログラミング機能の拡充（プラグインアクセス、ユーザアカウントの選択など）

■ 一連のデータベースユーザを符号化したBLOB内に保存

■ ログインおよびパスワードを入力してログインやパスワード満了規則を設定するための独自のダイアログボックス（最低文字数、定期的なパスワード変更要求など）

これらの変更のなかには、新規コマンドや既存コマンドの変更も含まれ、それぞれ後述しています。

注：アクセス管理システムにもインタフェースに関する若干の変更があります。新しいユーザ&グループエディタ、オンザフライでのパスワード変更機能、データベースアクセスの記録機能などです。

これらの新機能については、前述の「ログインダイアログボックス」の節で説明しています。

新しいコマンド

SET PLUGIN ACCESS

SET PLUGIN ACCESS (プラグイン; グループ)

引数	タイプ	説明
プラグイン	倍長整数	→ プラグイン番号
グループ	文字列	→ プラグインに関連付けるグループ名

SET PLUGIN ACCESS コマンドにより、データベースに組み込まれた“シリアル番号を持つ”各プラグインの使用を許可するユーザグループを、プログラムから設定することができます。

注：以前のバージョンの4th Dimensionにおいて、この操作は「デザイン」モードの「パスワード」メニューコマンドを使用して実行する以外の方法はありませんでした。

引数<プラグイン>には、ユーザグループを関連付けるプラグインの番号を渡します。プラグインライセンスには、WebやWebクライアントのライセンスが含まれます。「Is license available」テーマ内にあるいずれかの定数を渡すことができます。

定数	タイプ	値
Licence 4D Draw	倍長整数	808464694
Licence 4D for OCI	倍長整数	808465208
Licence 4D View	倍長整数	808465207
Licence 4D Web	倍長整数	808464945
Licence 4D Write	倍長整数	808464697
Licence Web Client	倍長整数	808465209

<グループ>には、そのプラグインの使用を許可するユーザが属すグループ名を渡します。

参照

Get plugin access

Get plugin access

Get plugin access(プラグイン) → 文字列

引数	タイプ		説明
プラグイン	倍長整数	→	プラグイン番号
戻り値	文字列	←	プラグインに関連付けられたグループ名

Get plugin access 関数は、引数<プラグイン>に渡した番号のプラグインの使用が許可されているユーザグループ名を返します。そのプラグインに関連付けられたグループが存在しない場合、コマンドは空の文字列 ("") を返します。

引数<プラグイン>には、ユーザグループが関連付けられたプラグインの番号を渡します。プラグインライセンスには、WebやWebクライアントのライセンスが含まれます。「Is license available」 テーマ内にあるいずれかの定数を渡すことができます。

定数	タイプ	値
Licence 4D Draw	倍長整数	808464694
Licence 4D for OCI	倍長整数	808465208
Licence 4D View	倍長整数	808465207
Licence 4D Web	倍長整数	808464945
Licence 4D Write	倍長整数	808464697
Licence Web Client	倍長整数	808465209

参照

SET PLUGIN ACCESS

USERS TO BLOB

USERS TO BLOB (ユーザ)

引数	タイプ		説明
ユーザ	BLOB	→	ユーザを格納する BLOB
		←	ユーザアカウント (符号化済)

USERS TO BLOB コマンドは、すべてのユーザアカウントのリストと、Administrator (管理者) が作成したデータベースグループを BLOB タイプの引数<ユーザ>に格納します。

このコマンドを実行できるのは、データベース管理者だけです。

生成された BLOB は自動的に符号化され、**BLOB TO USERS** コマンドを使用した場合にのみ読み取ることができます。この BLOB は、ハードディスク上のファイルやフィールドに保存可能です。

このコマンドは、ツールバーからグループやユーザを保存する場合と同等の働きをします。唯一異なる点は、このコマンドではファイルだけではなく、BLOB フィールドにもユーザアカウントを保存できるところです。

この概念を使用すると、各ユーザのバックアップをデータベースデータに保持することができます。したがって、バックアップメカニズムや、データベースストラクチャファイルの更新時にユーザを自動的にロードするシステムを設置することができます。ユーザアカウントに関連する情報は、4th Dimension によりデータベースストラクチャファイルに保存されます。

参照

BLOB TO USERS

BLOB TO USERS

BLOB TO USERS (ユーザ)

引数	タイプ	説明
ユーザ	BLOB	→ データベース管理者が作成し保存したデータベースのユーザアカウントを格納する BLOB (符号化済)

BLOB TO USERS コマンドは、BLOB タイプの引数<ユーザ>に格納されたユーザアカウントをデータベースへ追加します。BLOB<ユーザ>は符号化されており、**USERS TO BLOB** コマンドを用いて作成されていなければなりません。

このコマンドを実行できるのは、データベース管理者 (Administrator) だけです。他のユーザがこのコマンドを実行しようとする、コマンドは何も行わず、権限エラー (-9949) が生成されます。

ユーザアカウントを追加する際、次の規則が適用されます。

■ ユーザ ID は参照番号として使用されます。ユーザはそれぞれのユーザ ID に従って順番に処理されます。

- 番号がストラクチャファイル内に既に存在する場合、BLOBに格納された情報に従い、必要に応じてアカウント情報が変更されます。
- 番号がストラクチャファイル内に存在しない場合、BLOBに格納された情報に従い、ユーザが作成されます。
- その番号が、ストラクチャファイルで削除されたユーザアカウントと一致する場合、BLOBに格納された情報に応じてそのアカウントが変更されます。
- BLOBに格納された情報により、ユーザアカウントが削除されていることが判明した場合、そのアカウントはストラクチャファイルから削除されます。
- BLOBの情報に従って、更新後のユーザがグループに関連付けられます。
- グループが存在しなければ、作成されます。

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、それ以外の場合は0が代入されます。

互換性に関する注意：4th Dimension 2003の「パスワード」メニューにより作成されたユーザ&グループファイル(.4UG拡張子)は、次の手順を用いて4th Dimension 2004へロードすることができます。

DOCUMENT TO BLOB(mydoc; blob)

BLOB TO USERS(blob)

しかし、4th Dimension 2004で生成されたユーザ&グループファイルを4th Dimension 2003で開くことはできません。

参照

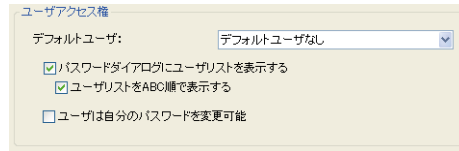
USERS TO BLOB

Get default user

Get default user → 数値

引数	タイプ	説明
戻り値	数値	← ユニークなユーザID番号

このコマンドは、データベースの「環境設定」ダイアログボックスで「デフォルトユーザ」として設定されたユーザのユニークなユーザIDを返します。



デフォルトユーザ

ユーザIDとして、次の数値が使用されます。

ID	ユーザの説明
1	Designer (デザイナー)
2	Administrator (管理者)
1 ~ 15000	Designerが作成したユーザ (ユーザ番号3はDesignerが作成した1番目のユーザ、ユーザ番号4は2番目のユーザなど)
-11 ~ -15010	Administratorが作成したユーザ (ユーザ番号-11はAdministratorが作成した1番目のユーザ、ユーザ番号-12は2番目のユーザなど)

デフォルトユーザが指定されていない場合、コマンドは0を返します。

変更されたコマンド

CHANGE CURRENT USER (CHANGE ACCESS)

CHANGE CURRENT USER {(ユーザ名 | ユーザ参照; パスワード)}

引数	タイプ	説明
ユーザ名 ユーザ参照	文字列 数値	→ ユーザ名またはユニークなユーザID
パスワード	文字列	→ パスワード (符号化されていない)

注：CHANGE CURRENT USERは、以前のバージョンの4th Dimensionで既に使用されていたCHANGE ACCESSコマンドの新しい名前です。明確にするため、このコマンドの名前が変更されました。

CHANGE CURRENT USER コマンドは、2つの任意の引数を受け入れるようになります。これらの引数を使用し、すべてのカレント“ユーザ”プロセスで使用される新しいアカウントをプログラムから指定することができます。他のプロセスで作成されたプロセスは、そのユーザアカウントを継承する点に留意してください。

4D Server：サーバマシン上で、「On Server Startup」、「On Server Shutdown」などのデータベースメソッドを用いて開始したプロセスは、デザイナー (Designer) アカウントのもとで実行されます。

1番目の引数として、使用するアカウントの<ユーザ名>またはユニークなユーザID (<ユーザ参照>) を渡します。ユーザの名前および番号は、**GET USER LIST** コマンドを用いて取得することができます。

ユーザIDとして、次の数値を使用できます。

ID	ユーザの説明
1	Designer (デザイナー)
2	Administrator (管理者)
1~15000	Designerが作成したユーザ (ユーザ番号3はDesignerが作成した1番目のユーザ、ユーザ番号4は2番目のユーザなど)
-11~-15010	Administratorが作成したユーザ (ユーザ番号-11はAdministratorが作成した1番目のユーザ、ユーザ番号-12は2番目のユーザなど)

ユーザアカウントが存在しない、または削除されている場合は、エラー-9979が返されません。このエラーは**ON ERR CALL** コマンドにより設定されたエラー処理用メソッドを用いて遮ることができます。

エラー処理メソッドを使用しない場合は、このコマンドを呼び出す前に、**Is user deleted** 関数を使用してユーザアカウントを検証することができます。

引数<パスワード>には、符号化されていないユーザアカウントパスワードを渡します。このパスワードがユーザと一致しない場合、コマンドはエラー-9978を返し、処理は行いません。

フラッディング (総当たり攻撃) つまり、複数のユーザ名/パスワードの組み合わせによる攻撃を避けるため、コマンドの実行は遅延されます。この結果、このコマンドは4度目に呼び出された後、10秒間待機した後ではじめて実行されます。この遅延はワークステーション全体にわたるものです。

注: **Validate password** コマンドもまた、無効な4回の呼び出し後に10秒間の遅延を行う方法を利用しています。

GET USER PROPERTIES

GET USER PROPERTIES (ユーザID; ユーザ名; 起動メソッド; パスワード; ログイン回数; 最終日付{; メンバー{; グループオーナー})

引数	タイプ	説明
ユーザID	数値	→ ユニークなユーザID番号
ユーザ名	文字列	← ユーザの名前
起動メソッド	文字列	← Startupメソッドの名前
パスワード	文字列	← 常に空の文字列
ログイン回数	数値	← データベースにログインした回数
最終日付	日付	← データベースに最後にログインした日付
メンバー	数値配列	← ユーザが属すグループのID番号
グループオーナー	数値	← ユーザグループオーナーのID番号

GET USER PROPERTIES コマンドは、グループの“オーナー”、つまりこのユーザが作成したオブジェクトのデフォルトグループオーナーのIDを任意の引数<グループオーナー>に返すようになりました。

Set user properties

Set user properties (ユーザID; ユーザ名; 起動メソッド; パスワード; ログイン回数; 最終日付{; メンバー{; グループオーナー}) → 数値

引数	タイプ	説明
ユーザID	数値	→ ユーザアカウントのユニークなID番号、またはDesignerに属するユーザを追加する場合は-1、またはAdministratorに属するユーザを追加する場合は-2
ユーザ名	文字列	→ 新しいユーザの名前
起動メソッド	文字列	→ 新しいユーザstartupメソッドの名前
パスワード	文字列	→ 新しい(符号化されていない)パスワード、またはパスワードを変更しない場合は“*”
ログイン回数	数値	→ データベースへの新しいログイン回数
最終日付	日付	→ データベースに最後にログインした新しい日付
メンバー	数値配列	→ ユーザが属すグループのID番号
グループオーナー	数値	← ユーザグループオーナーのID番号
戻り値	数値	← 新しいユーザのユニークなID番号

Set user properties 関数により、任意の引数<グループオーナー>を使用して、グループの“オーナー”、つまりこのユーザが作成したオブジェクトのデフォルトグループオーナーのIDを変更することができます。

独自のアクセス管理ダイアログボックスを提供する

「ユーザ&グループ」テーマの新しい機能を使用すると、ログインやパスワードの入力を行うために独自のダイアログボックスを実装することが可能であり、さらに、4th Dimensionの統合アクセスシステムも利用することができます。

例えば次のように、パスワードを制御できる高度な機能を設定できるようになります。

- 入力ルールを作成する（最小文字数、パスワードの一意性など）。
- 期限切れに関するルールを設定する（有効期限、初回ログイン時にパスワード変更を要求、数回のログイン失敗後にアカウントをロック、既に使用されたパスワードの保管など）。

これは以下の様に機能します。

- 1 データベースへのアクセスは、ダイアログボックスではなく、「デフォルトユーザ」を用いて直接行われる。
- 2 「On Startup」データベースメソッドにより、独自のユーザ名/パスワード入力用ダイアログボックスが表示される。

このメカニズムを利用することにより、あらゆる種類の動作を考えだすことができます。

- **GET USER LIST** コマンドを使用して、4th Dimensionのアクセスダイアログボックスのように、データベースユーザの一覧を表示することができます。
 - パスワードフィールドにさまざまな検証チェックを盛り込み、入力された文字が有効であることを確認することができます。
 - ダイアログボックスを確定した後は、期限に関するルールを適用することができます。つまり、初回接続、有効期限、無効なログインが繰り返された場合にアカウントをロックするなどのルールです。
- 3 入力が確定されると、ユーザアカウントのアクセス権を用いてデータベースを開くために必要な情報（ユーザ名とパスワード）が、**CHANGE CURRENT ACCESS** コマンドに渡される。

ウインドウ

4th Dimension 2004では、「ウインドウ」テーマのコマンドが機能強化され、さらに多くのウインドウサイズ調整用ツールが開発者に提供されています。また、新規コマンドが追加され（**Current form window** および **RESIZE FORM WINDOW**）、既存の **SET WINDOW RECT** コマンドの動作が変更されています。

デフォルトウインドウのサイズ調整管理が変更された点に注目してください。この件に関する詳細は、前述の「サイドプッシャー」の節を参照してください。

最後に、2つの新しいウインドウタイプが利用できます。

■ **Open window** コマンドではポップアップウインドウ（タイプ32）

■ **Open window** および **Open form window** 関数にはシートウインドウ（タイプ33）

新しいコマンド

Current form window

Current form window → ウインドウ参照番号

引数	タイプ	説明
----	-----	----

このコマンドには引数は必要ありません。

戻り値	ウインドウ参照	←	カレントフォームウインドウの参照番号
-----	---------	---	--------------------

このコマンドは、カレントフォームウインドウの参照を返します。カレントフォームに対してウインドウが定義されていない場合、コマンドは0を返します。

カレントフォームウインドウは、ユーザアクションの後、または **Open window** や **Open form window** 関数を使用することにより、**ADD RECORD** などのコマンドを用いて自動的に生成されます。

RESIZE FORM WINDOW

RESIZE FORM WINDOW (幅; 高さ)

引数	タイプ	説明
幅	倍長整数	→ カレントフォームウインドウの幅に追加、または減らすピクセル数
高さ	倍長整数	→ カレントフォームウインドウの高さに追加、または減らすピクセル数

このコマンドを使用し、カレント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。

引数<幅>と<高さ>には、カレントウインドウのサイズに加算したいピクセル数を渡します。カレントサイズを変更したくない場合は、どちらの引数にも0を渡します。サイズを小さくするには、引数<幅>と<高さ>に負の値を渡します。

このコマンドは、リサイズボックスを使用して手動でウインドウサイズを変更した場合（それが可能なウインドウタイプの場合）とまったく同じ結果をもたらします。結果として、このコマンドは、フォームプロパティで指定されたオブジェクトのサイズ調整プロパティやサイズ制限を考慮します。例えば、コマンドを用いて、フォームに許可されたサイズよりも大きなサイズにウインドウを変更すると、コマンドは何も行いません。

注目すべきなのは、この動作が**SET WINDOW RECT** コマンドの動作とは異なることです。**SET WINDOW RECT** コマンドは、ウインドウのリサイズを行う際に、フォームプロパティや内容を考慮しません。

4th Dimension 2004におけるこのコマンドの変更点に関する詳細は、後述の**SET WINDOW RECT** コマンドを参照してください。

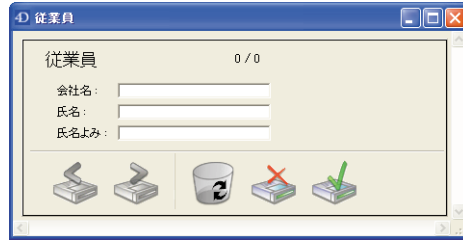
また、このコマンドは必ずしもフォームサイズを変更しないということに注意してください。プログラムからフォームサイズを変更するには、後述の**SET FORM SIZE** コマンドを参照してください。

▼ 次の図は、このコマンドの動作を表わしています。



これらのオブジェクトは、「水平サイジング」プロパティとして「拡大」が指定されている

RESIZE FORM WINDOW(25;0)



変更されたコマンド

SET WINDOW RECT

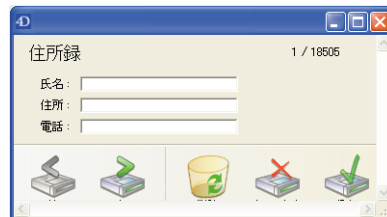
SET WINDOW RECT (左; 上; 右; 下; ウィンドウ)

4th Dimension 2004では、このコマンドの動作が変更されています。このコマンドはフォームオブジェクトに影響を与えなくなりました。

以前のバージョンの4th Dimensionでは、ウィンドウ内にフォームが含まれている場合、このコマンドによりフォームオブジェクトの移動やリサイズが行われていました(フォームプロパティに応じて)。今後、このコマンドではフォームやオブジェクトのプロパティが考慮されず、ウィンドウだけが変更されます。

サイズ調整プロパティが指定され、オブジェクトが存在するフォームウィンドウを更新するには、必ず前述の**RESIZE FORM WINDOW**コマンドを使用してください。

▼ 次の図は、このコマンドの動作を表しています。



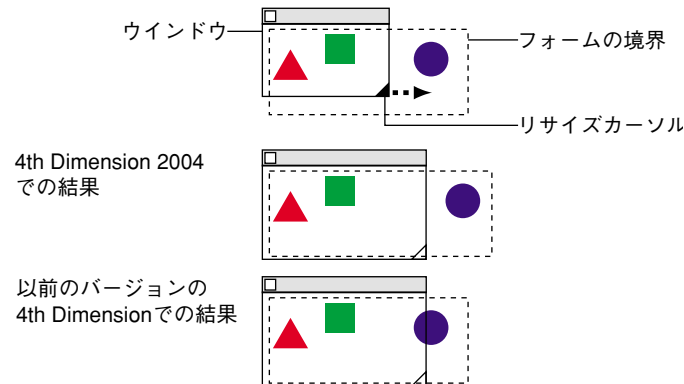
SET WINDOW RECT(100;100;300;300)



サイドブッシャー

4th Dimension 2004では、デフォルトとしてウインドウの右側および下側に“ブッシャー”スプリッタがあります。

つまり、ウインドウを大きくすると、画面上のウインドウの右側や下側の境界にあるオブジェクトが右方向や下方向へ自動的に押し出されます。

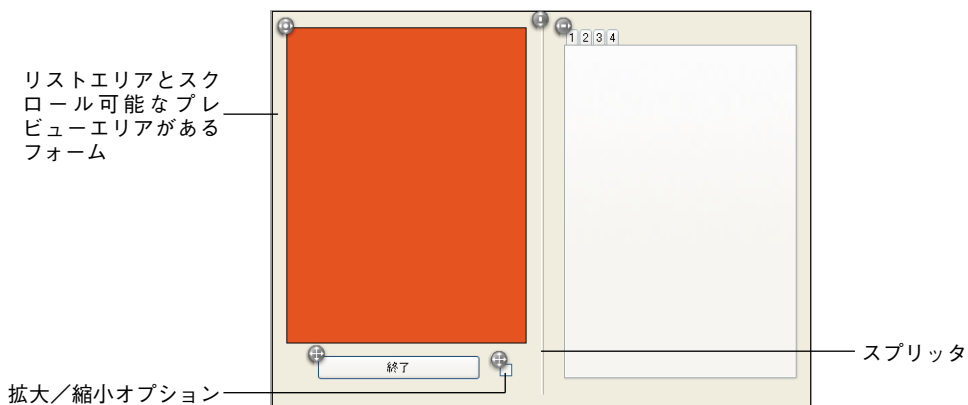


注：“ブッシャー”スプリッタに関する詳細は、前述の「スプリッタの管理」の節を参照してください。

ウインドウ管理機能と組み合わせると、このメカニズムにより、フォームを用いて「エクスプローラ」ウインドウのような格納式のウインドウを管理することができます（後述の例を参照）。

この機能は、スクロールバー付きのウインドウでは動作しません。

▼ 次の例題は、「エクスプローラ」タイプのウインドウの設定方法を示しています。以下のフォームは「デザイン」モードで作成します。

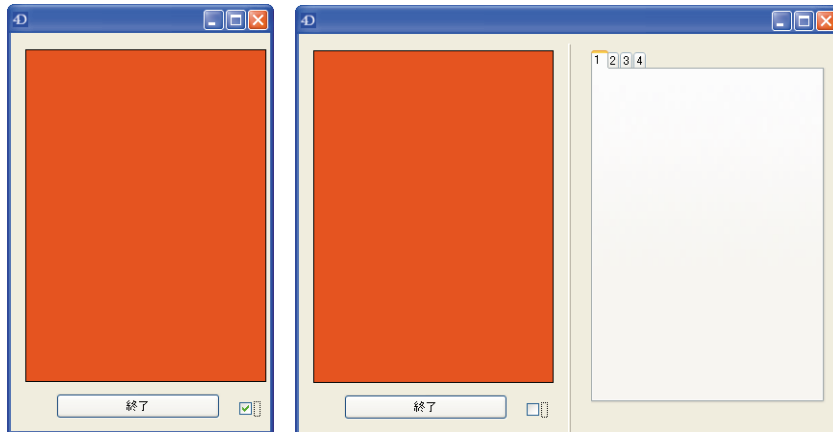


注：ウインドウサイズは“自動”です。

次のコードを使用して、ウインドウが表示されます。

```
$ref:=Open form window([Table 1];"Form1";Standard form window;  
Horizontally centered;Vertically centered;*)  
DIALOG([Table 1];"Form1")  
CLOSE WINDOW
```

このウインドウの右部分は、拡大/縮小ボックスをクリックすることにより、表示したり、または隠すことができます。



このオプションに関連付けられたオブジェクトメソッドは次の通りです。

Case of

¥ (Form event=On load)

C_BOOLEAN(b1;<>collapsed)

C_LONGINT(margin)

margin:=15

b1:=<>collapsed

If(<>collapsed)

SET FORM HORIZONTAL RESIZING((False)

SET FORM SIZE("b1";margin;margin)

Else

SET FORM HORIZONTAL RESIZING(True)

SET FORM SIZE("tab";margin;margin)

End if

¥ (Form event=On clicked)

<>collapsed:=b1

If(b1)

`縮小

GET OBJECT RECT(*;"b1";\$l;\$t;\$r;\$b)

GET WINDOW RECT(\$lf;\$tf;\$rf;\$bf;**Current form window**)

SET WINDOW RECT(\$lf;\$tf;\$lf+\$r+margin;

　　\$tf+\$b+margin;**Current form window**)

SET FORM HORIZONTAL RESIZING(False)

SET FORM SIZE("b1";margin;margin)

　　Else

　　`展開

GET OBJECT RECT(*;"tab";\$l;\$t;\$r;\$b)

GET WINDOW RECT(\$lf;\$tf;\$rf;\$bf;**Current form window**)

SET WINDOW RECT(\$lf;\$tf;\$lf+\$r+margin;

　　\$tf+\$b+margin;**Current form window**)

SET FORM HORIZONTAL RESIZING(True)

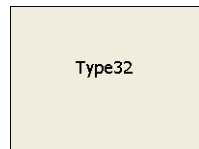
SET FORM SIZE("tab";margin;margin)

　　End if

End case

ポップアップウインドウ

「Open window」ならびに「Open form window」テーマには、新しい定数である **Pop up window (Pop up form window)** が追加されました (値は32)。タイプ32のウインドウを次に示します。



このタイプのウインドウは、タイプ2のウインドウの基本的な特性 (ウインドウタイトルあり) と次に示す高度な仕様を備えています。

- 以下の場合に、ウインドウは自動的に閉じられ、“キャンセル” イベントがウインドウに渡されます。
 - ウインドウの外側でクリックが行われた場合。
 - バックグラウンドウインドウまたはMDI (Multiple Document Interface) ウインドウが移動された場合。
 - ユーザがEscキーを押した場合。

ウインドウが表示される時、バックグラウンドウインドウは使用不可ではありませんが、イベントを受け取りません。

- マウスを使用して、ウインドウのリサイズや移動を行うことはできません。しかし、これらの動作をプログラムから実行すると、バックグラウンド項目のリドロウが最適化されます。

主としてこのタイプのウインドウは、3Dの“ベベル”や“ツールバー”ボタンに関連付けたポップアップメニューの処理に使用されます（前述の「新しい3Dボタン」の節を参照）。また、スブラッシュスクリーンの表示にも使用できます。

シートウインドウ

シートウインドウはMac OS X固有のもので、これらのウインドウは、アニメーションを用いてメインウインドウのタイトルバーの上に“ドロップダウン”され、メインウインドウの前面に表示されます。そのプロパティは、モーダルダイアログボックスのプロパティと同じです。通常、これらのウインドウは、メインウインドウで行われた動作に直接関連する動作を実行するために使われます。例えば、4th Dimensionでは、ユーザがラベルエディタの「読み込み」ボタンをクリックした時にシートウインドウが使用されています。

シートウインドウ



最後に開いたウインドウが表示されており、しかもドキュメントタイプ（フォーム）である場合にのみ、Mac OS X上でシートウインドウを作成することができます。

シートウインドウを管理するため、4th Dimensionに新しい定数が追加されました。

- 「Open window」 テーマ： Sheet window (33) および Resizable sheet window (34)
- 「Open form window」 テーマ： Sheet window (33)

Open window

Open window 関数に「Sheet window(33)」定数を渡すと、シートウインドウが開かれます。「Resizable sheet window(34)」を渡した場合は、シートウインドウのサイズを変更することができます。

以下の場合、コマンドはタイプ33のウインドウの代わりにタイプ1のウインドウを開き、タイプ34のウインドウの代わりにタイプ8のウインドウを開きます。

- 最後に開いたウインドウが表示されていないか、またはドキュメントタイプではない場合
- Windows上で実行された場合

Open form window

Open form window 関数に「Sheet window(33)」定数を渡すと、シートウインドウが開かれます。フォームが以下の条件を満たす場合、ウインドウのサイズを変更することができます。

以下の場合、コマンドはタイプ1のウインドウを開きます（または、サイズ変更可能であればタイプ8のウインドウ）。

- 最後に開いたウインドウが表示されていないか、またはドキュメントタイプではない場合
- Windows上で実行された場合

▼ シートウインドウはフォームの前面に描画しなければならないため、ウインドウでロードされた最初のフォームの「On load」イベントにおいて、その表示が押し戻されます。この例題は、表示の“遅延”を表わしています。

```
$myWindow:=Open form window([Table];"form";Sheet form window)
```

```
`シートフォームウインドウは作成されるが、非表示のままである
```

```
DIALOG([Table];"formDial")
```

```
`「On Load」イベントが生成され、次にシートウインドウが表示される
```

フォーム

新しいコマンドテーマ「フォーム」には、プログラムからのフォーム処理に使用するコマンドが用意されています。次のようなコマンドがあります。

- 4th Dimension 2004 の新しいフォーム管理用コマンド：**SET FORM SIZE**、**SET HORIZONTAL FORM RESIZING**、**SET FORM VERTICAL RESIZING**、**GET FORM OBJECTS**
- **INPUT FORM** コマンドと **OUTPUT FORM** コマンドの変更。今までこれらのコマンドは「テーブル」テーマに含まれていました。
- 既存のコマンドである **GET FORM PROPERTIES** コマンド。今までこのコマンドは「ウインドウ」テーマに含まれていました。コマンドの機能は変更されていません。
- 次のコマンドは今まで「フォームページ」テーマに含まれていました：**FIRST PAGE**、**LAST PAGE**、**NEXT PAGE**、**PREVIOUS PAGE**、**GOTO PAGE**、**Current form page**

SET FORM SIZE

SET FORM SIZE ({オブジェクト;} 水平; 垂直{;* })

引数	タイプ	説明
オブジェクト	文字列	→ フォームの境界を示すオブジェクト名
水平	倍長整数	→ “*” を渡した場合：水平マージン (ピクセル) “*” を省略した場合：幅 (ピクセル)
垂直	倍長整数	→ “*” を渡した場合：垂直マージン (ピクセル) “*” を省略した場合：高さ (ピクセル)
*	*	→ 渡した場合： ・自動サイズ調整、引数<水平>と<垂直>に指定されたマージンを追加、または ・オブジェクトに基づいたサイズ調整 (オブジェクトを渡した場合)、 引数<水平>と<垂直>に指定されたマージンを追加 省略した場合： ・フォームの幅および高さとして <水平>と<垂直>を使用する

このコマンドを使用すると、プログラムからカレントフォームのサイズを変更することができます。新しいサイズは、カレントプロセスに対して定義され、フォームと一緒に保存されません。

「デザイン」モードと同様に、このコマンドを使用して、3通りの方法でフォームサイズを設定することができます。

- 自動設定（すべてのオブジェクトを必ず表示するという考えに基づいてフォームサイズを決定）。“*” を渡した場合は、水平や垂直マージンを追加します。
- フォームオブジェクトが配置されていた位置に基づく。“*” を渡した場合は、この位置に水平や垂直マージンを追加します。
- “固定” サイズを入力（幅と高さ）。

フォームサイズの変更に関する詳細は、4th Dimension の『デザインリファレンス』マニュアルを参照してください。

自動サイズ調整

フォームサイズを自動的に設定したい場合、必ず次の構文を使用してください。

SET FORM SIZE (水平; 垂直;*)

この場合、フォームの右側と下側に追加したいマージン（ピクセル単位）を<水平>と<垂直>に代入して渡します。

オブジェクトに基づいたサイズ調整

任意のオブジェクトに基づいてフォームサイズを設定したい場合、必ず次の構文を使用してください。

SET FORM SIZE (オブジェクト; 水平; 垂直;*)

この場合、オブジェクトの右側と下側に追加したいマージン（ピクセル単位）を<水平>と<垂直>に代入して渡します。

固定サイズ

フォームの固定サイズを設定したい場合、必ず次の構文を使用してください。

SET FORM SIZE (水平; 垂直)

この場合、フォームの幅と高さ（ピクセル単位）を<水平>と<垂直>に代入して渡します。

SET FORM SIZE コマンドはフォームサイズを変更しますが、サイズ調整プロパティも考慮します。例えば、フォームの最小幅が500ピクセルの場合に、コマンドで幅を400ピクセルに設定すると、新しいフォーム幅は500ピクセルになります。

また、このコマンドはフォームウインドウのサイズは変更しないという点に注意してください。ウインドウサイズを変更しないでフォームサイズを変えることもできます（その逆も同様）。フォームウインドウのサイズを変更する方法については、前述の **RESIZE FORM WINDOW** コマンドを参照してください。

SET FORM HORIZONTAL RESIZING

SET FORM HORIZONTAL RESIZING (リサイズ{; 最小幅{; 最大幅})

引数	タイプ	説明
リサイズ	ブール	→ True (真) : フォームは水平方向にリサイズ可能 False (偽) : フォームは水平方向にリサイズできない
最小幅	倍長整数	→ 許容される最小フォーム幅 (ピクセル)
最大幅	倍長整数	→ 許容される最大フォーム幅 (ピクセル)

SET FORM HORIZONTAL RESIZING コマンドを使用すると、プログラムからカレントフォームの水平リサイズプロパティを変更することができます。デフォルトとして、これらのプロパティは「デザイン」モードの「フォーム」エディタにおいて設定することができます。新しいプロパティはカレントプロセスに対して設定され、フォームと一緒に保存されません。

引数<リサイズ>を用いて、フォームを水平方向にリサイズできるかどうか、つまり、幅の変更が可能かどうかを定義します（ユーザが手動で変更、またはプログラムから変更）。

<True>を渡すと、ユーザはフォームの幅を変更することができます。4th Dimension はマーカとして<最小幅>と<最大幅>に代入された値を使用します。

<False>を渡すと、ユーザはカレントフォームの幅を変更できません。この場合、引数<最小幅>と<最大幅>に値を渡す必要はありません。

1番目の引数に<True>を渡した場合は、任意の引数<最小幅>と<最大幅>に最小幅と最大幅の新しい値（ピクセル単位）を渡すことができます。これらの引数を省略した場合、「デザイン」モードで設定した値（設定されている場合）が使用されます。

SET FORM VERTICAL RESIZING

SET FORM VERTICAL RESIZING (リサイズ{; 最小高さ{; 最大高さ})

引数	タイプ	説明
リサイズ	ブール	→ True (真) : フォームは垂直方向にリサイズ可能 False (偽) : フォームは垂直方向にリサイズできない
最小高さ	倍長整数	→ 許容されるフォームの最小の高さ (ピクセル)
最大高さ	倍長整数	→ 許容されるフォームの最大の高さ (ピクセル)

SET FORM VERTICAL RESIZING コマンドを使用すると、プログラムからカレントフォームの垂直リサイズプロパティを変更することができます。デフォルトとして、これらのプロパティは「デザイン」モードの「フォーム」エディタにおいて設定することができます。新しいプロパティはカレントプロセスに対して設定され、フォームと一緒に保存されません。

引数<リサイズ>を用いて、フォームを垂直方向にリサイズできるかどうか、つまり、高さの変更が可能かどうかを定義します (ユーザが手動で変更、またはプログラムから変更)。

<True>を渡すと、ユーザはカレントフォームの高さを変更することができます。4th Dimension はマーカとして<最小高さ>と<最大高さ>に代入された値を使用します。

<False>を渡すと、ユーザはカレントフォームの高さを変更できません。この場合、引数<最小高さ>と<最大高さ>に値を渡す必要はありません。

1番目の引数に<True>を渡した場合は、任意の引数<最小高さ>と<最大高さ>に最小の高さと最大の高さの新しい値 (ピクセル単位) を渡すことができます。これらの引数を省略した場合、「デザイン」モードで設定した値 (設定されている場合) が使用されません。

GET FORM OBJECTS

GET FORM OBJECTS (オブジェクト配列{; 変数配列{; ページ配列}}; *)

引数	タイプ	説明
オブジェクト配列	文字列配列	← フォームオブジェクト名
変数配列	ポインタ配列	← オブジェクトに関連付けた変数やフィールドへのポインタ
ページ配列	整数配列	← 各オブジェクトのページ番号
*	*	→ 渡した場合：カレントページに限定する

GET FORM OBJECTS コマンドは、カレントテーブルのカレントフォームに存在する全オブジェクトのリストを配列形式で返します。このリストは、カレントフォームページのオブジェクトに限定することができます。このコマンドは、入力フォームおよび出力フォームの双方で使用することができます。

引数として渡した配列が事前に定義されていない場合、コマンドはその配列を作成し、サイズを自動的に設定します。しかし、アプリケーションをコンパイルする場合を考慮し、各配列を明示的に宣言することをお勧めします。

オブジェクト名を格納する文字列配列の名前（フォーム内の各オブジェクトはユニークな名前を持つ）を<オブジェクト配列>に渡します。配列内でのオブジェクトの出現順序は重要ではありません。

コマンドにより代入される他の任意の配列は、1番目の配列との同期が取られます。

任意の引数<変数配列>にはポインタ配列の名前を渡し、この配列にはオブジェクトに関連付けられている変数やフィールドへのポインタが格納されます。オブジェクトに関連付けられた変数が存在しない場合、ポインタ“Nil”が返されます。“サブフォーム”オブジェクトタイプが存在する場合、サブフォームのテーブルへのポインタが返されます。

3番目の配列（任意）<ページ配列>には、フォームのページ番号が代入されます。この配列の各要素には、対応するオブジェクトのページ番号が格納されます。

継承フォームによってもたらされるオブジェクトは、カレントページのページ0に属しているものとみなされます。

任意の引数<*>を使用すると、返されるオブジェクトのリストをフォームのカレントページに限定することができます。この引数を渡した場合、コマンドはカレントページ、ページ0、継承ページのオブジェクトだけを返します。このコマンドは、フォームのカレントページ上に存在するあらゆるオブジェクトを処理します。

変更されたコマンド

次のコマンドは、今まで「テーブル」テーマに属していました。

INPUT FORM

INPUT FORM ({テーブル;}フォーム{; ユーザフォーム}{; *})

引数	タイプ	説明
テーブル	テーブル	→ 入力フォームが設定されたテーブル、またはこの引数を省略した場合はデフォルトテーブル
フォーム	文字列	→ 使用するフォーム名
ユーザフォーム	文字列	→ 使用されるユーザフォーム名
*	*	→ 自動ウインドウサイズ調整

INPUT FORM コマンドは新たに任意の引数<ユーザフォーム>を受け入れるようになりました。この引数を使用して、ユーザフォームをデフォルトの入力フォームとして設定することができます。

注：ユーザフォームに関する詳細は、前述の「ユーザフォーム」の節を参照してください。

引数<ユーザフォーム>に有効なユーザフォーム名を渡すと、カレントプロセスにおいてデフォルトとしてこのフォームが使用されます。

これにより、カスタマイズされた各種ユーザフォーム (**CREATE USER FORM** コマンドで生成されたフォーム) を同時に使用できるようになり、現在の状況で一番効果を発揮するフォームを使用することができます。

▼ 複数の企業を管理する請求書データベースにおいて、対応するユーザフォームで請求書を作成しなくてはなりません。

Case of

```
¥ (company="4D SA")
```

```
INPUT FORM([Invoices];"Entry";"4D_SA")
```

```
¥ (company="4D Inc")
```

```
INPUT FORM([Invoices];"Entry";"4D_Inc")
```

```
¥ (company="Acme")
```

```
INPUT FORM([Invoices];"Entry";"ACME")
```

End case

```
ADD RECORD([Invoices])
```

参照

OUTPUT FORM

OUTPUT FORM

OUTPUT FORM ({テーブル;}フォーム{; ユーザフォーム})

引数	タイプ	説明
テーブル	テーブル	→ 出力フォームが設定されたテーブル、またはこの引数を省略した場合はデフォルトテーブル
フォーム	文字列	→ 使用するフォーム名
ユーザフォーム	文字列	→ 使用されるユーザフォーム名

OUTPUT FORM コマンドは新たに任意の引数<ユーザフォーム>を受け入れるようになりました。この引数を使用して、ユーザフォームをデフォルトの出力フォームとして設定することができます。

注：ユーザフォームに関する詳細は、前述の「ユーザフォーム」の節を参照してください。

引数<ユーザフォーム>に有効なユーザフォーム名を渡すと、カレントプロセスにおいてデフォルトとしてこのフォームが使用されます。

これにより、カスタマイズされた各種ユーザフォーム (**CREATE USER FORM** コマンドで生成されたフォーム) を同時に使用できるようになり、現在の状況で一番効果を発揮するフォームを使用できます。

リレート

4th Dimension 2004 では、プログラムを用いてリレートをより強力に制御できるようになりました。

■ 新しいコマンド、**GET FIELD RELATION** および **SET FIELD RELATION** を使用して、各リレートのステータスを個別に設定することができます。

■ 新しい **GET AUTOMATIC RELATIONS** コマンドにより、データベースにおける各リレートの現在のステータスを調べることができます。

注：一貫性を保つため、既存のコマンドである **AUTOMATIC RELATIONS** コマンドの名前が **SET AUTOMATIC RELATIONS** に変更されています。その機能は変わりません。

GET FIELD RELATION

GET FIELD RELATION (始点フィールド; 1対1; 1対n; *)

引数	タイプ	説明
始点フィールド	フィールド	→ リレートの開始点となるフィールド
1対1	倍長整数	← n対1のリレートのステータス
1対n	倍長整数	← 1対nのリレートのステータス
*	*	→ 渡した場合：<1対1>と<1対n>にはリレートのカレントステータスが返される（値“2”または“3”のみ） 省略した場合（デフォルト）：プログラムによりリレートが変更されていなければ、<1対1>と<1対n>には値“1”が返される

GET FIELD RELATION コマンドを使用すると、カレントプロセスの<始点フィールド>から開始するリレートの自動/マニュアルのステータスを調べることができます。

「ストラクチャ」ウィンドウで設定した自動リレートをはじめとして、あらゆるリレートを調べることができます。

■ <始点フィールド>には、ステータスを調べようとするリレートが開始するnテーブルのフィールド名を渡します。

フィールド<始点フィールド>から開始するリレートが存在しない場合、引数<1対1>と<1対n>には0が返されます。また、エラーが返されて、システム変数OKには0が代入されます（後述）。

■ コマンドの実行後、引数<1対1>には、指定したn対1リレートが自動的に設定されているかどうかを示す値が格納されます。

■ 0=<始点フィールド>から始まるリレートが存在しません。シンタックスエラー番号16（“このフィールドにはリレートが設定されていません。”）が生成され、システム変数OKには0が代入されます。

■ 1=指定されたn対1リレートの自動/マニュアルのステータスは、「デザイン」モードの「リレートプロパティ」内の「自動1対1リレート」オプションにより設定されたものです（プログラムにより変更されていない）。

■ 2=そのプロセスのn対1のリレートはマニュアルです。

■ 3=そのプロセスのn対1のリレートは自動です。

■ コマンドの実行後、引数<1対n>には、指定した1対nリレートが自動として設定されているかどうかを示す値が格納されます。

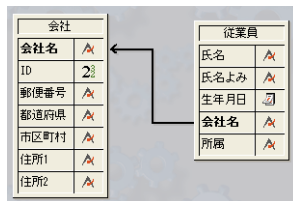
- 0=引数<始点フィールド>から始まるリレートが存在しません。シンタックスエラー番号16 (“このフィールドにはリレートが設定されていません。”) が生成され、システム変数OKには0が代入されます。
- 1=指定された1対nリレートの自動/マニュアルのステータスは、「デザイン」モードの「リレートプロパティ」にある「自動1対nリレート」オプションにより設定されたものです（プログラムにより変更されていない）。
- 2=そのプロセスの1対nのリレートはマニュアルです。
- 3=そのプロセスの1対nのリレートは自動です。

引数<1対1>および<1対n>に返された値は、「リレート」テーマの定数と比較することができます。

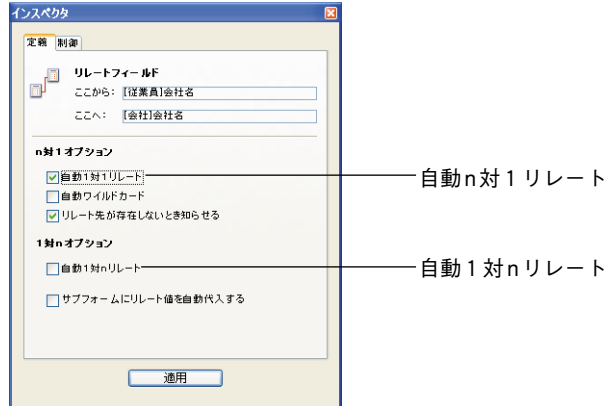
定数	タイプ	値
No relation	倍長整数	0
Structure configuration	倍長整数	1
Manual	倍長整数	2
Automatic	倍長整数	3

- 任意の引数<*>を使用すると、プログラムから修正されていない場合でも、リレートのカレントステータスを“強制的に”読み込むことができます。言い換えれば、任意の引数<*>を渡した場合、引数<1対1>および<1対n>には値“2”または“3”だけが返されます。

▼ 次のようなストラクチャについて検証しましょう。



[従業員]会社フィールドから[会社]名前フィールドへリンクするリレートのプロパティは次の通りです。



次のコードは、**GET FIELD RELATION** コマンドおよび**SET FIELD RELATION** コマンドにより提供されるさまざまな機能とともに、その効果を示します。

```
C_BOOLEAN($is_one;$is_many)
```

```
C_LONGINT(one;many)
```

```
GET AUTOMATIC RELATIONS($is_one;$is_many) `False, False を返す
```

```
GET FIELD RELATION([従業員]会社名;one;many) `1、1 を返す
```

```
GET FIELD RELATION([従業員]会社名;one;many;*) `3、2 を返す
```

```
SET FIELD RELATION([従業員]会社名;2;0)
```

　`n 対 1 リレートをマニュアルに変更する

```
GET FIELD RELATION([従業員]会社名;one;many) `2、1 を返す
```

```
GET FIELD RELATION([従業員]会社名;one;many;*) `2、2 を返す
```

```
SET FIELD RELATION([従業員]会社名;1;0)
```

　`n 対 1 リレートに対し

　`「デザイン」モードで設定された引数を再設定する

```
GET FIELD RELATION([従業員]会社名;one;many) `1、1 を返す
```

```
GET FIELD RELATION([従業員]会社名;one;many;*) `3、2 を返す
```

```
SET AUTOMATIC RELATIONS(True;True) `全テーブルのすべてのリレートを
```

自動に変更する

```
GET AUTOMATIC RELATIONS($is_one;$is_many) `True、True を返す
```

```
GET FIELD RELATION([従業員]会社名;one;many) `1、1 を返す
```

```
GET FIELD RELATION([従業員]会社名;one;many;*) `3、3 を返す
```

▼ 次は、**GET FIELD RELATION** と **SET FIELD RELATION** の2つのコマンドの代表的な使用方法を示す例題です。

```
C_LONGINT(one,many)
```

```

`現在の引数を読み込む
GET FIELD RELATION([MyTable]MyField;one;many)
`n対1リレーをマニュアルに変更し、1対nリレーは自動のままにする
SET FIELD RELATION([MyTable]MyField;3;2)
QUERY(...) `クエリの実行
`元の引数の再設定
SET FIELD RELATION([MyTable]MyField;one;many)

```

参照

SET FIELD RELATION

SET FIELD RELATION

SET FIELD RELATION (nテーブル | 始点フィールド; 1対1; 1対n)

引数	タイプ	説明
nテーブル; 始点フィールド	テーブル;フィールド →	リレーの開始点となるテーブル、 またはリレーの開始点となる フィールド
1対1	倍長整数;ブール →	フィールドから開始するn対1の リレーのステータス、または テーブルのn対1のリレー
1対n	倍長整数;ブール →	フィールドから開始する1対nの リレーのステータス、または テーブルの1対nのリレー

SET FIELD RELATION コマンドを使用すると、「デザイン」モードの「リレープロパティ」ウインドウで設定した初期ステータスが何であれ、データベースの各リレーの自動/マニュアルのステータスを個別に設定することができます。

■ 1番目の引数には、テーブルやフィールドの名前を渡します。

■ フィールド名 (<始点フィールド>) を渡すと、コマンドは、指定した始点フィールドから開始するリレーに対してのみ適用されます。

■ テーブル名 (<nテーブル>) を渡すと、コマンドは指定したnテーブルから開始するリレーに対してのみ適用されます。

■ <始点フィールド>または<nテーブル>から開始するリレーが存在しない場合、引数<1対1>と<1対n>には0が返されます。シンタックスエラー番号16 (“このフィールドにはリレーが設定されていません。”) が生成されて、システム変数OKには0が代入されます。

- 引数<1対1>には、指定したn対1リレートに適用される自動/マニュアルのステータス変更を示す値を渡します。この値として、「Relations」テーマの定数を使用できません。
- Do not modify(0) = n対1リレートの現在のステータスを変更しない。
- Structure configuration(1) = アプリケーションのストラクチャウインドウで指定されたn対1リレートの設定を使用する。
- Manual(2) = カレントプロセスに対し、n対1リレートをマニュアルに設定する。
- Automatic(3) = カレントプロセスに対し、n対1リレートを自動に設定する。
- 引数<1対n>には、指定した1対nリレートに適用される自動/マニュアルのステータス変更を示す値を渡します。この値として、「Relations」テーマの定数を使用できません。
- Do not modify(0) = 1対nリレートの現在のステータスを変更しない。
- Structure configuration(1) = アプリケーションのストラクチャウインドウで指定された1対nリレートの設定を使用する。
- Manual(2) = カレントプロセスに対し、1対nリレートをマニュアルに設定する。
- Automatic(3) = カレントプロセスに対し、1対nリレートを自動に設定する。

注：このコマンドを使用して行った変更は、カレントプロセスに対してのみ適用されません。「リレートプロパティ」ウインドウのオプションを用いて指定されたリレート設定は変更されません。

- ▼ この新しいコマンドを使用することにより、クイックレポートのリレート管理がさらに容易になります。以前のバージョンの4th Dimensionでは、エディタ上でリレートを利用するには、すべてのリレートを自動に設定する必要がありました。次のコードを使用すると、有用なリレートだけを自動に設定できるようになります。

```
SET AUTOMATIC RELATIONS(False;False) `リレートのリセット
`次のリレートだけが使用される
SET FIELD RELATION([Invoices]ID_Cus;Automatic;Automatic)
SET FIELD RELATION([Invoice_row]ID_Invoice;Automatic;Automatic)
QR REPORT([Invoices];Char(1);True;True;True)
```

GET AUTOMATIC RELATIONS

GET AUTOMATIC RELATIONS (1対1; n)

引数	タイプ	説明
1対1	ブール	← すべてのn対1リレーのステータス
nブール	←	すべての1対nリレーのステータス

GET AUTOMATIC RELATIONS コマンドにより、データベースで設定されたすべてのマニュアルn対1リレーおよび1対nリレーに関する自動/マニュアルのステータスがカレントプロセスにおいて変更されたかどうかを知ることができます。このコマンドは、**AUTOMATIC RELATIONS** コマンド（4th Dimension 2004では**SET AUTOMATIC RELATIONS**へ名称変更）の逆のコマンドです。

■ 1対1：前回の**SET AUTOMATIC RELATIONS** コマンドの呼び出しにより、すべてのマニュアルn対1リレーが自動的に設定された場合、この引数は“True”を返します（例えば**SET AUTOMATIC RELATIONS(True;False)**）。

また、**SET AUTOMATIC RELATIONS** コマンドがコールされなかった場合や、前回の実行によりマニュアルn対1リレーが変更されなかった場合（例えば**SET AUTOMATIC RELATIONS(False;False)**）には、“False”を返します。

■ 1対n：前回の**SET AUTOMATIC RELATIONS** コマンドの呼び出しにより、すべてのマニュアル1対nリレーが自動的に設定された場合、この引数は“True”を返します（例えば**SET AUTOMATIC RELATIONS(True;True)**）。

また、**SET AUTOMATIC RELATIONS** コマンドがコールされなかった場合や、前回の実行によりマニュアル1対nリレーが変更されなかった場合（例えば**SET AUTOMATIC RELATIONS(True;False)**）は、“False”を返します。

▼ **GET FIELD RELATION** コマンドの例題を参照してください。

フォームイベント

4th Dimension 2004では、新しいフォームイベントが提供されています。これらの新しいフォームイベントは、次のニーズに応えるものです。

- ロールオーバー管理
- 3D ボタンの管理
- 新しい“リストボックス”オブジェクトの管理
- リスト上のレコードの管理
- 階層リストの管理

ロールオーバーに関連するフォームイベント

新しい3つのフォームイベントにより、あらゆるフォームオブジェクトに関し、マウスを用いてロールオーバーを処理することができます。

- On Mouse Enter
- On Mouse Move
- On Mouse Leave

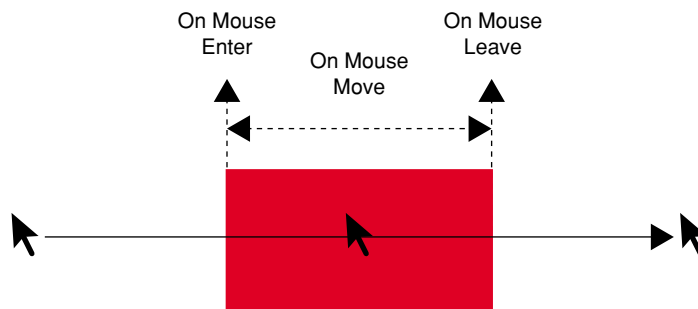
以前のバージョンの4th Dimensionでは、ロールオーバーの管理には複雑なメソッドを作成する必要がありました。4th Dimension 2004では、「On Mouse Enter」、「On Mouse Move」、「On Mouse Leave」イベントを使用することにより、ロールオーバーを簡単に管理することができます。

これらのイベントは、インタフェースを備えたアクティブウィンドウに対し、任意のプロセスで生成されます。2つの重なり合ったオブジェクトが存在する場合、それを完全に管理できる1番目のオブジェクトによって、イベントが生成されます。

SET VISIBLE コマンドで非表示に設定されたオブジェクトは、これらのイベントを生成しません。あるオブジェクトが入力されると、他のオブジェクトはマウスの位置に応じてロールオーバーイベントを受け取ることができます。

フォームにおいてこれらのイベントを選択すると、各フォームオブジェクトに対してそれぞれのイベントが生成されます。

- オブジェクトの画像エリアにマウスを移動すると、「On Mouse Enter」イベントが発生します。オブジェクトメソッドは一度呼び出されます。
- オブジェクトの画像エリア内でマウスを移動するたびに（少なくとも1ピクセル）、「On Mouse Move」が発生します。オブジェクトメソッドはそのたびに呼び出されます。
- マウスがオブジェクトの画像エリアから離れると、「On Mouse Leave」イベントが発生します。オブジェクトメソッドは一度呼び出されます。



3D ボタンに関連するフォームイベント

4th Dimension 2004 には、新しい3D ボタンファミリーが導入され、高度なグラフィックインタフェースを設置することができます（新しい3D ボタンに関する詳細は、前述の「新しい3D ボタン」の節を参照）。

これらの新しいオブジェクトを管理するため、新たに2つのフォームイベント、「On long click」および「On arrow click」が作成されました。

■ **On long click**：このイベントは、3D ボタンがクリックを受け取り、マウスボタンが一定の時間押したままにされた場合に生成されます。理論上、このイベントが生成される時間の長さは、システム環境設定で定義したダブルクリックを区切る最長時間と同じです。

このイベントは、“前世代”の3D ボタン（つまり、バックグラウンドオフセット）やポップアップメニューが付属する3D ボタンの矢印エリアを除き、あらゆるタイプの3D ボタン、3D ラジオボタン、3D チェックボックスに対して生成されます。

通常、このイベントは、ボタンクリックを長く行った時にポップアップメニューを表示するために使用されます。ユーザが“長いクリック”のタイムアウト前にマウスボタンを離すと、「On click」イベントが有効であれば、それが生成されます。

■ **On arrow click**：3D ボタンのなかには、ポップアップメニューに関連付けて、矢印を表示できるものがあります。その矢印をクリックすると、選択用のポップアップメニューが表示され、最初のボタンアクションに関連する追加アクションを設定することができます。



矢印付きの3D ボタン例（“ツールバーボタン”スタイル）

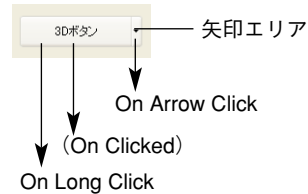
4th Dimension では、新しいフォームイベントである「On arrow click」を用いて、このタイプのボタンを管理することができます。ユーザが“矢印”をクリックすると、このイベントが生成されます（マウスボタンが押された直後）。

■ ポップアップメニューが“分離”タイプの場合、矢印のあるボタン部分でクリックが行われた場合にのみ、このイベントが生成されます。

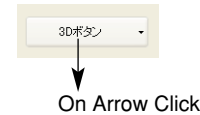
■ ポップアップメニューが“リンク”タイプの場合、ボタンの任意の場所でクリックが行われると、このイベントが生成されます。このタイプのボタンには、「On long click」イベントが生成されないということに注意してください。

3D ボタン例 (“ツールバーボタン” スタイル)

分離型のポップアップメニューの場合



リンク型のポップアップメニューの場合



次の3Dボタン、3Dラジオボタン、3Dチェックボックスボタンスタイルは、「ポップアップメニュー」プロパティに対応します。

- なし
- ツールバーボタン
- ベベル
- 角の丸いベベル
- Office XP

リストボックスに関連するフォームイベント

リストボックスは4th Dimension 2004に取り入れられた新しいオブジェクトタイプです。「フォーム」エディタにおけるリストボックスタイプのオブジェクトの作成や設定に関する詳細は、「デザインモード」の節を参照してください。また、プログラムでの管理方法については、後述の「リストボックス」の節を参照してください。

次の7つの新しいフォームイベントを使用して、リストボックスの特性を取り扱うことができます：**On Before Data Entry**、**On Selection Change**、**On Column Moved**、**On Row Moved**、**On Column Resize**、**On Header Click**、**On After Sort**。

これらのイベントは、リストボックスオブジェクトに対して使用可能であり、さまざまな状況を管理します。

- **On Before Data Entry**：このイベントは、リストボックスのセルが編集される直前に生成されます（入力カーソルが表示される前）。例えば、このイベントを使用すると、開発者はユーザが参照中か、編集中かによって異なるテキストを表示することができます。
- **On Selection Change**：このイベントは、リストボックスの行や列のカレント選択項目が変更されるたびに生成されます。

注：また、このイベントはレコードリスト（後述の「レコードリストに関連するフォームイベント」の節を参照）や、階層リスト（後述の「階層リストに関連するフォームイベント」の節を参照）においても生成されます。

■ **On Column Moved**：このイベントは、ユーザがドラッグ&ドロップを使い、リストボックスの列を移動すると生成されます。ドラッグ&ドロップで列を最初の位置に戻した場合には、このイベントは生成されません。

MOVED LISTBOX COLUMN NUMBER コマンドは、列の新しい位置を返します。

■ **On Row Moved**：このイベントは、ユーザがドラッグ&ドロップを使い、リストボックスの行を移動すると生成されます。ドラッグ&ドロップで行を最初の位置に戻した場合には、このイベントは生成されません。

MOVED LISTBOX ROW NUMBER コマンドは、行の新しい位置を返します。

■ **On Column Resize**：このイベントは、リストボックスのカラム幅が変更されると生成されます（マウスまたはプログラム上で**SET LISTBOX COLUMN WIDTH**コマンドを使用）。

■ **On Header Click**：このイベントは、リストボックスの列ヘッダ上でクリックが行われると生成されます。

注：この場合、Self関数を使用してクリックされた列ヘッダを知ることができます（後述の「Self」の節を参照）。

リストボックスにおいて、「並び替え可」プロパティが選択されている場合、変数\$0に値“0”や“-1”を渡すことにより、標準的なカラムソートを許可するかどうかを決定することができます。

■ \$0が0の場合、標準の並び替えが実行されます。

■ \$0が-1の場合、標準の並び替えは実行されず、ヘッダには並び替えを示す矢印が表示されません。この場合でも、開発者は4th Dimensionの配列管理コマンドを使用して、独自の並び替え条件に基づいて列のソートを行うことができます。

リストボックスにおいて、「並び替え可」プロパティが選択されていない場合、変数\$0は使用されません。

■ **On After Sort**：このイベントは、標準の並び替えが実行された直後に生成されます（ただし、「On Header Click」イベントで\$0に-1が返された場合、このイベントは生成されません）。このメカニズムは、ユーザが前回実行した並び替えの方向を保存する際に役立ちます。

このイベントにおいて、Selfコマンドは並び替えが行われた列の変数へのポインタを返します。

注：リストボックスを使用する際、列や列ヘッダ上で右クリック（Mac OS上ではCtrl+クリック）を行うと、「On Clicked」イベントが生成されます。

レコードリストに関連するフォームイベント

4th Dimension 2004では、リストとして表示されたレコード（「ユーザ」モードで**MODIFY SELECTION**、**DISPLAY SELECTION**コマンドを使用、またはサブフォーム内）の管理が変更されました。すべての新機能については、前述の「レコードリストの管理」の節で説明しています。

新しい2つのフォームイベント、「On selection change」と「On Load Record」を使用すると、これらレコードの表示処理がさらに向上します。

■ **On Selection Change**：リストフォームやサブフォーム内のカレントレコードまたは選択項目が変更されるたびに、このイベントが生成されます。

注：また、このイベントは、リストボックスオブジェクト（前述の「リストボックスに関連するフォームイベント」の節を参照）や、階層リスト（後述の「階層リストに関連するフォームイベント」の節を参照）においても生成されます。

■ **On Load Record**：このイベントは、「リスト更新」モードにおいて、編集可能な状態（ユーザがレコードの行をクリックし、フィールドが入力可になる）でレコードがロードされると生成されます。

On Display Detail/ On Header

「On Display Detail」および「On Header」イベントから、ダイアログボックスを表示する4Dコマンドを呼び出せなくなりました。これを行った場合は、シンタックスエラーが生成されるようになります。

特にこの件に関連するコマンドは、次の通りです：**ALERT**、**DIALOG**、**CONFIRM**、**Request**、**ADD RECORD**、**MODIFY RECORD**、**DISPLAY SELECTION**、**MODIFY SELECTION**。

階層リストに関連するフォームイベント

4th Dimension 2004では、いくつかのコマンドが変更されたため、階層リストをより楽に管理できるようになりました（前述の「階層リスト」の節を参照）。

また、新しい3つのフォームイベント、「On selection change」、「On expand」、「On collaps」を使用すると、階層リスト上で実行されたユーザアクションの処理がさらに向上します。

■ **On selection change** : マウスクリックやキーストロークの後、階層リストの選択項目が変更されるたびに、このイベントが生成されます。

注：また、このイベントは、リストボックスオブジェクト（前述の「リストボックスに関連するフォームイベント」の節を参照）や、レコードリスト（前述の「階層リストに関連するフォームイベント」の節を参照）においても生成されます。

■ **On expand** : マウスクリックやキーストロークを用いて階層リストの項目が展開されるたびに、このイベントが生成されます。

■ **On Collapse** : マウスクリックやキーストロークを用いて階層リストの項目が縮められるたびに、このイベントが生成されます。

これらのイベントは、相互に排他的ではありません。階層リストに対し、連続して生成される場合があります。

キーストローク後（順に） :

イベント	コンテキスト
On data change	項目が編集された
On expand/ On collapse	→や←の矢印キーを用いたサブリストのオープン／クローズ
On selection change	新しい項目の選択
On clicked	キーボードを用いたリストのアクティブ化

マウスクリック後（順に） :

イベント	コンテキスト
On data change	項目が編集された
On expand/ On collapse	展開／縮小アイコンを用いたサブリストのオープン／クローズ または、編集不可のサブリスト上でダブルクリック
On selection change	新しい項目の選択
On clicked/ On double clicked	クリックやダブルクリックを用いたリストのアクティブ化

ユーザインタフェース

次の節では、新しいコマンドである **GET TABLE TITLES**、**GET FIELD TITLES**、**SCROLL LINES** の説明、ならびに **Pop up menu** 関数と **Focus object** 関数（今までの **Last object**）の変更点について説明します。

GET TABLE TITLES

GET TABLE TITLES (テーブルタイトル; テーブル番号)

引数	タイプ	説明
テーブルタイトル	文字列配列	← カレントテーブル名
テーブル番号	数値配列	← テーブル番号

GET TABLE TITLES コマンドは、ストラクチャウインドウや **SET TABLE TITLES** コマンドを用いて指定された各データベーステーブルの名前と番号を、配列<テーブルタイトル>と<テーブル番号>に代入します。これら2つの配列内容は同期化されます。

そのセッション中に、**SET TABLE TITLES** コマンドが呼び出されると、**GET TABLE TITLES** コマンドは、そのコマンドを用いて設定された“変更後”の名前とテーブル番号だけを返します。

それ以外の場合、**GET TABLE TITLES** コマンドはストラクチャウインドウで定義されたすべてのデータベーステーブルの名前を返します。

両方の場合ともに、コマンドは非表示テーブルを返しません。

参照

GET FIELD TITLES、SET TABLE TITLES

GET FIELD TITLES

GET FIELD TITLES (テーブル; フィールドタイトル; フィールド番号)

引数	タイプ	説明
テーブル	テーブル	→ その名前を調べたいフィールドが属すテーブル
フィールドタイトル	文字列配列	← 現在のフィールド名
フィールド番号	数値配列	← フィールド番号

GET FIELD TITLES コマンドは、目的の<テーブル>のフィールド名およびフィールド番号を配列<フィールドタイトル>と<フィールド番号>に代入します。これら2つの配列内容は同期化されます。

そのセッション中に **SET TABLE TITLES** コマンドが呼び出されると、**GET FIELD TITLES** コマンドは、そのコマンドを用いて設定された“変更後”の名前とフィールド番号だけを返します。

それ以外の場合、**GET FIELD TITLES** コマンドはストラクチャウインドウで定義されたすべてのデータベースフィールドの名前を返します。

両方の場合ともに、コマンドは非表示フィールドを返しません。

参照

GET TABLE TITLES、SET FIELD TITLES

SCROLL LINES

SCROLL LINES ({*; }オブジェクト{; 位置}{; * })

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>はテーブル
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、またはテーブル（“*”を省略した場合）
位置	倍長整数	→ セレクション中のレコードの位置
*		→ スクロール後に行を最初の位置に表示する

SCROLL LINES コマンドを使用すると、最初に選択したレコードや特定のレコードが表示されるように、フォームリスト（**MODIFY SELECTION**、**DISPLAY SELECTION** コマンドを用いて表示される）またはサブフォームの行をスクロールすることができます。

最初の任意の引数“*”を渡すと、引数<オブジェクト>がサブフォームオブジェクトの名前であることを示します（この場合、<オブジェクト>には文字列を渡す）。この引数を渡さない場合、引数<オブジェクト>がテーブルであることを示します（リストフォームテーブルまたはサブフォームテーブル）。

引数<位置>を渡さない場合、リスト内で最初に選択したレコード（強調表示される）が表示されるように、リスト行を縦方向にスクロールします。

選択されたレコードが存在しない場合、コマンドは何も行いません。少なくとも1つの選択レコードが表示されている場合も、コマンドは何も行いません。

引数<位置>を使用すると、現在の選択項目中のレコード番号を指示することができます（その位置）。この引数を渡すと、指定したレコード（強調表示されているかどうかに関わらず）が表示されるように、リスト行が縦方向にスクロールされます。レコードが既に表示されている場合、コマンドは何も行いません。

2番目の任意の引数“*”を渡すと、このコマンドを用いて表示される行はリストの一番目の位置に配置されます。

注：HIGHLIGHT RECORDS コマンドには引数 “*”（任意）が導入され、これを使用するとスクロール管理を SCROLL LINES コマンドに委ねることができます。

参照

HIGHLIGHT RECORDS

Pop up menu

Pop up menu (項目テキスト{; デフォルト{; X座標; Y座標}) → 数値

引数	タイプ	説明
項目テキスト	テキスト	→ メニューテキスト定義
デフォルト	数値	→ デフォルトとして選択されるメニュー項目の番号
X座標	数値	→ 左上端のX座標
Y座標	数値	→ 左上端のY座標
戻り値	数値	← 選択されたメニュー項目番号

Pop up menu コマンドは2つの任意の引数を受け入れ、これらの引数を用いて、表示しようとするポップアップメニューの左上端の位置を正確に指定することができます。

<X座標>と<Y座標>にはそれぞれ、メニューの左上端の水平座標と垂直座標を渡します。カレントフォームのローカルな座標システムにおいて、これらの座標をピクセル単位で表わさなくてはなりません。必ずこれら2つの引数は一緒に渡してください。一方だけを渡しても、引数は無視されます。

引数<X座標>と<Y座標>を使用する場合は、引数<デフォルト>が渡されても無視されます。この場合、ポップアップメニューにマウスを配置する必要はありません。

これらの新しい引数は、ポップアップメニューが関連付けられた3Dボタンを管理する上で特に役立ちます。

Focus object

注：Focus object は、既存の Last object 関数の新しい名前です。明確にするため、このコマンドの名前が変更されました。

Focus object → ポインタ

Focus object 関数はカレントフォーム内でフォーカスを取得しているオブジェクトへのポインタを返します。フォーカスを取得したオブジェクトが存在しない場合、コマンドは Nil を返します。

リストボックスタイプのオブジェクトに対して使用すると、**Focus object**関数はフォーカスを取得したリストボックスの列（配列）へのポインタを返します。詳細については、後述の「フォーカスオブジェクト」の節を参照してください。

入力制御

このテーマには新しく **EDIT ITEM** コマンドが納められ、このコマンドによりリストタイプのオブジェクト（階層リスト、リストボックス、サブフォーム）を編集モードに移行することができます。

EDIT ITEM

EDIT ITEM ({*; } オブジェクト{; 項目})

引数	タイプ	説明
*	*	→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>はフィールドまたは変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、またはフィールドまたは変数（“*”を省略した場合）
項目	数値	→ 項目番号

このコマンドを使用すると、引数<オブジェクト>に指定したリストや配列において、カレント項目または引数<項目>で指定した番号の項目を編集することができます。つまり、選択した項目は変更され、入力された文字で項目内容がすべて置き換えられます。

任意の引数“*”を渡すと、引数<オブジェクト>がオブジェクト名であることを示します（この場合、<オブジェクト>には文字列を渡す）。この引数を渡さない場合、引数<オブジェクト>がフィールド、または変数であることを示します。この場合、文字列ではなく、フィールドや変数の参照番号を渡します。

このコマンドは次の入力可オブジェクトに対して適用されます。

■ 階層リスト

■ リストボックス（後述の「リストボックス」の節を参照）

■ サブフォーム（この場合、<オブジェクト>にはオブジェクト名（サブフォーム）だけを渡すことが可能）

■ MODIFY SELECTIONや**DISPLAY SELECTION** コマンドで表示されたリストフォーム

リストではない入力可オブジェクトに対してこのコマンドを使用すると、**GOTO AREA** コマンドと同じ動作を行います。

リストや配列が空であるか非表示の場合、コマンドは何も行いません。また、リストや配列が入力不可の場合、コマンドは指定された項目を選択するだけで、編集モードには移行しません。

任意の引数<項目>を使用すると、編集モードに変更する項目の位置（階層リスト）または行番号（リストボックス、リストフォーム、“複数選択”モードのサブフォーム）を指定することができます。この引数を渡さない場合、コマンドは<オブジェクト>のカレント項目に対して適用されます。

カレント項目が存在しなければ、<オブジェクト>の1番目の項目を編集モードに変更します。

注：・階層リストにおいて、EDIT ITEM コマンドは自動的にリストをリドロウします。したがって、EDIT ITEM コマンドを使用する場合は、REDRAW LIST コマンドを呼び出すべきではありません。

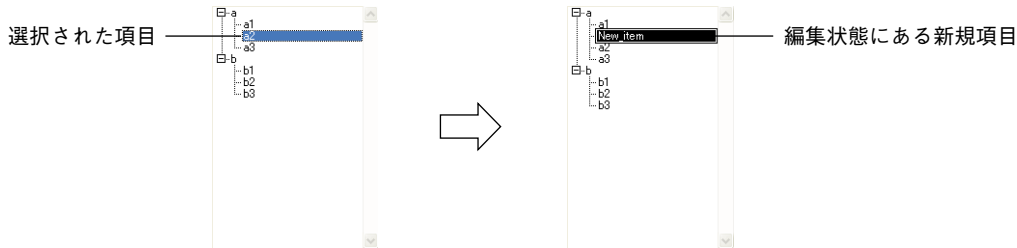
・サブフォームとリストフォームにおいて、入力を行えるように、コマンドは指定された行の最初のフィールドを入力モードに変更します。

・“1レコード”選択モードのサブフォームに関しては、引数<項目>の値に関わらず、コマンドはカレント行を入力可にします。

▼ このコマンドは、階層リストに新しい項目を作成する際に特に役立ちます。コマンドが呼び出されると、リストへ追加または挿入された最後の項目が自動的に入力可能となり、ユーザによる動作は必要ありません。

次のコードは、既存のリストに新規項目を挿入するボタンのメソッドとして使用することができます。デフォルトとして表示される“New_item”というテキストは、自動的に変更を受け入れられる状態になります。

```
vlUniqueRef:=vlUniqueRef+1
INSERT LIST ITEM(hList;*;"New_item";vlUniqueRef)
EDIT ITEM(*;"MyList")
```



- ▼ リストボックスに2つのカラムが存在し、それぞれの変数名は“配列1”と“配列2”であるものとします。次の例題は、リストボックスに新しい行を挿入し、配列2の新規項目を強調表示します。

```

$vlRowNum:=Size of array(xsNames)+1
INSERT LISTBOX ROW(*,"MyListbox";$vlRowNum)
xsNames{$vlRowNum}:="新しい値 1"
xsAdr1{$vlRowNum}:="新しい値 2"
EDIT ITEM(xsAdr1;$vlRowNum)

```

氏名	都道府県
梅野和久	東京都
梅若明雄	神奈川県
梶川雄	千葉県

氏名	都道府県
梅野和久	東京都
梅若明雄	神奈川県
梶川雄	千葉県
新しい値1	新しい値2

- ▼ 次の例題を使用して、サブセレクション中の最終サブレコードの1番目のフィールドを編集モードに変更することができます。

```

LAST SUBRECORD([Children])
EDIT ITEM(*,"Subform")

```

印刷

サーバ上での印刷

大部分の4th Dimension印刷コマンドは、ストアドプロシージャとして4D Server上で実行できるようになりました。例えば、サーバマシン上でPDFドキュメントを印刷する際に、この機能が役立ちます。

次の印刷コマンドを使用できます。

■ PRINT SELECTION

■ PRINT RECORD

■ PRINT LABEL

■ QR REPORT

■ Print form

印刷用の引数を設定するコマンド（つまり、**SET PRINT OPTION**）も利用することができます。

印刷コマンドを使用する上で、次の情報が非常に役立ちます。

■ サーバマシン上にはダイアログボックスを一切表示しないでください（特定の必要性がある場合を除く）。

これを行うには、**PRINT SELECTION**、**PRINT RECORD**、**PRINT LABEL**、**QR REPORT**の各コマンドを呼び出す際に、引数<*>を指定しなくてはなりません。これにより、印刷ダイアログボックスが表示されなくなります。

■ 4D Serverでは、エディタ（ラベルエディタおよびクイックレポートエディタ）の呼び出しに使用する構文は動作しません。これを行った場合、システム変数OKの値は0に設定されます。

■ 印刷時に問題が発生しても（用紙切れ、プリンタ停止など）、4th Dimensionはエラーメッセージを生成しません。

SET PRINT OPTION/GET PRINT OPTION（「印刷」テーマ）

SET PRINT OPTION (オプション; 値1{; 値2})

GET PRINT OPTION (オプション; 値1{; 値2})

注：4th Dimension 2003.3において、この新しい機能が導入されました。

Mac OS X上では、デフォルトとして印刷はPDFとして行われます。PDFのプリンタドライバでは、カプセル化PostScript情報（EPS）を含むPICTピクチャがサポートされません。例えば、これらのピクチャはベクトル描画ソフトにより生成されます。

この問題を回避するため、4th Dimensionでは新しい引数<オプション>を利用できます。この引数により、Mac OS X上でカレントセッション中に使用する印刷モードを設定することができます。デフォルトとして、印刷ジョブはPDFで行われますが（標準モード）、開発者は必要に応じて**SET PRINT OPTION**と**GET PRINT OPTION**コマンドを使用し、この設定を変更することができます。

■ Mac spool file format option(13)：印刷ジョブをPDFモード（デフォルト値）に設定する場合は0を渡し、印刷ジョブを“強制的に”PostScriptモード（カプセル化PostScriptを含むPICTの印刷）に設定する場合は1を渡します。

また、このオプションにより、4D Write、4D View、4D Drawプラグインの印刷モードを変更することができます。

これに対し、Windows上においてこのコマンドは無効です。

重要：バージョン2003.3より以前の4th Dimensionにおいて、印刷ジョブは“強制的に”PostScriptモードに設定されていました。今後、モード0がデフォルトとして使用されます。以前のモードを維持したい場合（特に、カプセル化PostScriptを含むPICTがデータベースに存在する場合）、値“1”を用いてこのオプションを呼び出さなくてはなりません。

メッセージ

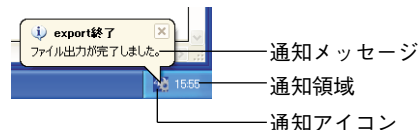
DISPLAY NOTIFICATION

DISPLAY NOTIFICATION (タイトル; テキスト{; 表示時間})

引数	タイプ	説明
タイトル	文字 255 →	メッセージタイトル
テキスト	文字 255 →	メッセージテキスト
表示時間	数値	→ 表示を継続する時間（秒単位）

注：このコマンドはWindows上でのみ動作します。

DISPLAY NOTIFICATION コマンドは、Windowsのタスクバーの通知領域にメッセージを表示します。



通常、このタイプのメッセージは、OSやアプリケーションがユーザに外部イベント（ネットワーク切断、アップグレードの提供）を知らせるために使用されます。

<タイトル>と<テキスト>には、表示するメッセージのタイトルとテキストを渡します（上記の例では、タイトルは“export終了”）。255桁までの文字を入力することができます。

デフォルトでは、ユーザがクローズボックスをクリックするまで、メッセージウインドウは表示されたままです。任意の引数<表示時間>を渡すと、ユーザがクローズボックスをクリックしなかった場合、指定した継続時間に達した時点でウインドウが自動的に閉じられます。ユーザがウインドウをクローズした場合でも、<表示時間>が終了するまで通知アイコンは表示されたままになります。

データ入力

DIALOG

DIALOG ({テーブル;}フォーム)

4th Dimension 2004では、<フォーム>にボタンが存在しない場合に、「OK」ボタンや「キャンセル」ボタンが追加されなくなります。この場合、フォームを終了するためには、Escapeだけを使用できます。

注：4th Dimension 2004では、このコマンドにより表示されたフォームのフィールドに値を入力できるようになりました。「環境設定」ダイアログボックスの新しいオプションを使用して、この動作を変更することができます（前述の「新しい互換性オプション」の節を参照）。

システムドキュメント

Select document

Select document (ディレクトリ;ファイルタイプ;タイトル; オプション{; 選択項目})→文字列

引数	タイプ	説明
ディレクトリ	テキスト; 倍長整数 →	・ドキュメント選択ダイアログボックスにデフォルトとして表示するディレクトリのアクセスパス、または ・デフォルトのユーザフォルダを表示する場合は空の文字列 (Windowsでは“My documents”、Mac OSでは“Documents”)、または ・保存されたアクセスパスの番号
ファイルタイプ	テキスト →	ふり分けするドキュメントタイプの一覧、またはドキュメントのふり分けを行わない場合は “*”
タイトル	テキスト →	選択ダイアログボックスのタイトル
オプション	倍長整数 →	選択オプション
選択項目	テキスト配列 ←	選択したファイルのアクセスパスと名前の一覧を格納する配列
戻り値	文字列 ←	選択したファイルの名前 (複数項目が選択された場合は1番目のファイル)

Select document 関数は、標準の「ドキュメントを開く」ダイアログボックスを表示し、これを用いてユーザは1つ以上のファイルを指定し、その選択したファイルの名前やフルアクセスパスを返すことができます。

引数<ディレクトリ>はフォルダを表わし、最初にそのフォルダの内容がドキュメントを開くダイアログボックス上に表示されます。3種類の値を渡すことができます。

■ 表示するフォルダのフルアクセスパスを含むテキスト

■ 現行のオペレーティングシステムのデフォルトユーザフォルダを表示する場合は、空の文字列 (“”) (Windowsでは“My documents”、Mac OSでは“Documents”)

■ 関連するフォルダを表示するために保存されたアクセスパスの番号 (1～32000)

したがって、ユーザが選択ボタンをクリックした際に開かれるフォルダのアクセスパス、つまり、ユーザにより選択されたフォルダをメモリ上に保存することができます。最初に任意の番号を呼び出すと（例えば、“5”）、コマンドはそのオペレーティングシステムのデフォルトユーザフォルダを表示します（空の文字列を渡す場合と同じ）。また、ユーザはハードディスク上のフォルダを閲覧することもできます。ユーザが選択ボタンをクリックすると、アクセスパスが保存されて番号“5”に関連付けられます。この後で番号“5”を呼び出すと、保存されたアクセスパスがデフォルトとして使用されます。新しい場所が選択された場合は、パス番号5が更新されます。

このメカニズムにより、32,000個までのアクセスパスを記憶しておくことができます。Windows上では、各パスはそのセッションでのみ保持されます。Mac OS上では、各パスはシステムによって保持され、セッションからセッションへと保存されます。

引数<ファイルタイプ>には、ファイルを開くダイアログボックス上で選択できるファイルタイプを渡します。複数のタイプを“;（セミコロン）”で区切ったリストを渡すことができます。定義された各タイプごとに、行がダイアログボックスのタイプ選択メニューに追加されます。

Mac OS上では、ふるい分けられるファイルのタイプを渡さなくてはなりません。標準のOSTypeフォーマット（TEXT、APPL）や（Mac OS 10.3以降を使用している場合は）新しいUTIフォーマット（public.jpeg、public.tiff）を使用することができます。また、両方のフォーマットを引数<ファイルタイプ>に混在させることもできます。

Windows上で、Mac OSタイプのファイル（4th Dimensionが内部的に変換を行います）やファイルの拡張子（.txt、.exe）を渡すこともできます。Windowsでは、ダイアログボックスに“*.*”と入力すると、すべてのドキュメントタイプを“強制的”に表示できる点に注意してください。しかしこの場合、4th Dimensionは選択されるファイルタイプの追加検証を実行します。ユーザが許可されていないタイプを選択すると、コマンドはエラーを返します。

表示されるファイルをいくつかのタイプに限定したくない場合、<ファイルタイプ>に値“*（アスタリスク）”または“.*”を渡します。

引数<タイトル>には、ダイアログボックスに表示しなければならないラベルを渡します。空の文字列を渡すと、“Open”というラベルが表示されます。

引数<オプション>を使用し、ファイルを開くダイアログボックスで許可される上級機能を指定することができます。4th Dimensionでは、次に示す「System documents」テーマの定義済み定数が用意されています。

定数	タイプ	値
Multiple files	倍長整数	1
Package open	倍長整数	2
Package selection	倍長整数	4
Alias selection	倍長整数	8

定数のうちの一つを渡すか、あるいは定数を組み合わせて渡すことができます。

■ **Multiple files** : キーコンビネーションの Shift+クリック (隣接する項目の選択)、Ctrl+クリック (Windows)、Command+クリック (Mac OS) を使用し、複数ファイルを同時に選択することを許可します。この場合、引数<選択項目>を渡すと、選択されたファイルの一覧がこの引数に代入されます。

この定数を使用しない場合、デフォルトとしてコマンドは複数ファイルの選択を許可しません。

■ **Package open (Mac OSのみ)** : パッケージのオープンならびにその内容の表示を許可します。

この定数を使用しない場合、デフォルトとしてコマンドはパッケージのオープンを許可しません。

■ **Package selection (Mac OSのみ)** : 実体としてパッケージの選択を許可します。

この定数を使用しない場合、デフォルトとしてコマンドはソフトウェアパッケージをそのまま使用することを許可しません。この場合、ソフトウェアパッケージのオープンや選択は行えません (定数「Package open」を渡した場合でも)。

■ **Alias selection** : ショートカット (Windows) やエイリアス (Mac OS) をドキュメントとして選択することを許可します。

デフォルトでは、この定数を使用しない場合にショートカットやエイリアスを選択すると、コマンドは目的とする項目のアクセスパスを返します。

■ **Use Sheet Window (Mac OSのみ)** : シートウインドウとして選択用ダイアログボックスを表示します (Windowsでは、このオプションは無視)。

シートウインドウはMac OS X固有のインタフェースで、グラフィックアニメーションを利用しています (詳細については、前述の「シートウインドウ」の節を参照)。

この定数が使用されない場合、デフォルトとしてコマンドは標準のダイアログボックスを表示します。

定数を渡した場合、コマンドはエイリアスやショートカット自体のアクセスパスを返します。

任意の引数<選択項目>を使用して、ユーザが選択した各ファイルのフルアクセスパス（アクセスパス+ファイルの名前）を取得することができます。コマンドは、ユーザの選択に応じて配列の作成、サイズの調整、値の代入を行います。この引数は、「Multiple files」を選択した場合や、選択したファイルへのアクセスパスを調べたい場合に役立ちます（コマンドにより返された配列の値からファイル名を取り除く）。

ファイルが選択されない場合、コマンドは空の配列を返します。

コマンドは選択されたファイルの名前（Windowsでは名前+拡張子）を返します。

複数のファイルが選択された場合、コマンドは選択ファイルリスト内の最初のファイル名を返します。このファイル一覧は、任意の引数<選択項目>により取得することができます。

ファイルが選択されない場合、コマンドは空の文字列を返します。

コマンドが正常に実行され、有効なドキュメントが選択されると、システム変数OKには1が代入され、システム変数Documentには選択されたファイルのフルアクセスパスが代入されます。

ファイルが選択されない場合（例えば、ファイルを開くダイアログボックスでユーザが「キャンセル」ボタンをクリックした場合）、システム変数OKには0が代入され、システム変数Documentは空白になります。

オブジェクトプロパティ

- 「オブジェクトプロパティ」テーマが拡張されて、新しいコマンド **Get format** が加えられました。このコマンドを使用すると、各オブジェクトの現在の表示フォーマットを知ることができます。
- また、既存のコマンド、**SET ENTERABLE**、**SET FORMAT**、**SET RGB COLORS**、**SET COLOR** が機能強化されています。

Get format

Get format ({*;} オブジェクト) → 文字列

引数	タイプ	説明
*	*	→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>はフィールドまたは変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 またはフィールドまたは変数（“*”を省略した場合）
戻り値	文字列	← オブジェクトの表示フォーマット

Get format関数は、引数<オブジェクト>で指定されたオブジェクトに適用された現在の表示フォーマットを返します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名であることを示します（この場合<オブジェクト>には文字列を渡す）。引数<*>を渡さない場合、引数<オブジェクト>がフィールドまたは変数であることを示します。その場合、文字列ではなく、フィールドや変数の参照番号を渡してください。

注：関数を一連のオブジェクトに対して適用した場合、最後に選択されたオブジェクトのフォーマットが返されます。

この関数は、オブジェクトの現在の表示フォーマット、つまり、「デザイン」モードや**SET FORMAT**コマンドで定義されたフォーマットを返します。**Get format**関数は、表示フォーマットを受け入れるあらゆるタイプのフォームオブジェクト（フィールドや変数）に対して使用することができます（つまり、プル、日付、時間、ピクチャ、文字列、数値、ボタングリッド、ダイヤル、サーモメータ、ルーラ、ピクチャポップアップメニュー、ピクチャボタン、3Dボタン）。これらのオブジェクトの表示フォーマットに関する詳細は、**SET FORMAT**コマンドの説明を参照してください。

Get format関数を日付、時間、ピクチャタイプのオブジェクト（定数として定義されたフォーマット）に対して適用すると、定数のASCIIコードに相当する文字列が返されます。定数の値を取得するには、この戻り値に対して**Ascii**関数を適用してください（後述）。

▼この例題により、“myphoto”という名前のピクチャ変数に対して適用されたフォーマット定数の値を取得することができます。

```
C_STRING(2;$format)
SET FORMAT(*;"myphoto";Char(On background))
  `バックグラウンドフォーマットを適用（値=3）
$format:=Get format(*;"myphoto")
```

```
ALERT("フォーマット番号："+String(Ascii($format)))
`値 “3” を表示
```

▼この例題を使用し、ブールタイプのフィールド “[メンバー]婚姻区分” に対して適用されたフォーマットを取得することができます。

```
C_STRING(30;$format)
$format:=Get format([メンバー]婚姻区分)
ALERT($format)
`例えば “既婚：独身” のようなフォーマットを表示。
```

SET ENTERABLE

SET ENTERABLE ({*; }入力エリア; 入力可)

引数	タイプ	説明
*	*	→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>はテーブル、フィールド、変数のいずれか
入力エリア	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、またはテーブル、フィールドまたは変数（“*”を省略した場合）
入力可	ブール	→ True=入力可、False=入力不可

SET ENTERABLE コマンドを使用すると、**MODIFY SELECTION** や **DISPLAY SELECTION** コマンドを用いて表示されたリストフォーム、およびサブフォームに対して、「リスト更新」オプションをプログラムから有効にすることができます。

■ リストフォームの場合、例えば **SET ENTERABLE**([MyTable];True) のように、引数<オブジェクト>にはフォームのテーブル名を渡さなくてはなりません。

■ サブフォームの場合、例えば **SET ENTERABLE**(*;"Subform";True) のように、サブフォームのテーブル名またはサブフォームオブジェクト名を引数<オブジェクト>に渡すことができます。

注：MODIFY SELECTION および DISPLAY SELECTION コマンドの構文も変更され、開発者がプログラムからリストフォームをより管理しやすくなっています。

▼次は、リストのヘッダに配置されたチェックボックスのオブジェクトメソッドであり、リスト更新モードを処理します。

```
C_BOOLEAN(bEnterable)
SET ENTERABLE([Table1];bEnterable)
```

SET FORMAT

SET FORMAT ({*;} オブジェクト; 表示フォーマット)

引数	タイプ	説明
*	*	→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>はフィールドまたは変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、またはフィールド または変数（“*”を省略した場合）
表示フォーマット	数値	→ オブジェクトの新しい表示フォーマット

SET FORMAT コマンド（ならびに、その逆のコマンドである **Get format**）は、特に 3D ボタンを表示するために設計されています（3D ボタン、3D ラジオボタン、3D チェックボックス）。

注：4th Dimension 2004 において、3D ボタンはいくつかの新しいオプションを受け入れるようになりました。詳細については、前述した「新しい 3D ボタン」の節を参照してください。

3D ボタン

3D ボタンにフォーマットを適用するには、次のシンタックスを用いて <表示フォーマット> に文字列を渡します。

タイトル;ピクチャ;バックグラウンド;タイトル位置;タイトル表示;アイコン表示;スタイル;水平マージン;垂直マージン;アイコンオフセット;ポップアップメニュー

■ タイトル=ボタンタイトル

この値は、テキストまたはリソース番号として表わすことができます（例：“:16800,1”）。

■ ピクチャ=ボタンに関連付けたピクチャ。ピクチャライブラリ、ピクチャ変数、または PICT リソースから参照する。

- ・ピクチャライブラリから参照するピクチャの場合、疑問符記号に続けて番号を入力します（例：“?250”）。
- ・ピクチャ変数から参照するピクチャの場合、変数名を入力します。
- ・PICT リソースから参照するピクチャの場合、コロンに続けて番号を入力します（例：“:62500”）。

- **バックグラウンド**=ボタンに関連付けたバックグラウンドピクチャ（カスタムスタイル）。ピクチャライブラリ、ピクチャ変数、またはPICTリソースから参照する（上記参照）。
- **タイトル位置**=ボタンタイトルの位置。5つの値を指定できます。
 - ・タイトル位置=0：中央
 - ・タイトル位置=1：右
 - ・タイトル位置=2：左
 - ・タイトル位置=3：下
 - ・タイトル位置=4：上
- **タイトル表示**=タイトルを表示するかどうかを指定します。2つの値を指定できます。
 - ・タイトル表示=0：タイトルを隠す
 - ・タイトル表示=1：タイトルを表示する
- **アイコン表示**=アイコンを表示するかどうかを指定します。2つの値を指定できます。
 - ・アイコン表示=0：アイコンを隠す
 - ・アイコン表示=1：アイコンを表示する
- **スタイル**=ボタンのスタイル。このオプションの値により、他のさまざまなオプションを考慮するかどうかが決まります（例えば、バックグラウンド）。10種類の値を指定できます。
 - ・スタイル=0：なし
 - ・スタイル=1：バックグラウンドオフセット
 - ・スタイル=2：プッシュボタン
 - ・スタイル=3：ツールバーボタン
 - ・スタイル=4：カスタム
 - ・スタイル=5：円
 - ・スタイル=6：システムスクエア小
 - ・スタイル=7：Office XP
 - ・スタイル=8：ベベル
 - ・スタイル=9：角の丸いベベル
- **水平マージン**=水平マージン。

ボタンの左マージンと右マージンの内側境界を定めるピクセル数（アイコンとテキストを配置してはならない領域）。

■ 垂直マージン=垂直マージン。

ボタンの上マージンと下マージンの内側境界を定めるピクセル数（アイコンとテキストを配置してはならない領域）。

■ アイコンオフセット=右方向または下方向へのアイコンの移動。

この値はピクセル単位で表わされ、ボタンのクリック時に、ボタンアイコンを右方向および下方向へ移動することを指示します（両方向ともに同じ値が使用されます）。

■ ポップアップメニュー=ポップアップメニューとボタンとの組み合わせ。3つの値を指定できます。

- ・ポップアップメニュー=0：ポップアップメニューなし
- ・ポップアップメニュー=1：リンクしたポップアップメニューあり
- ・ポップアップメニュー=2：分離したポップアップメニューあり

これらのオプションのなかには、すべての3Dボタンスタイルに使用できないものもあります。また、状況によっては、すべてのオプションを変更したくない場合もあります。任意のオプションを渡さないようにするには、対応する値を省略してください。例えば、「タイトル表示」と「垂直マージン」オプションを渡さないようにするには、次のように記述します。

```
SET FORMAT(myVar;"NiceButton"?256;;562;1;;1;4;5;;5;0")
```

SET RGB COLORS

SET RGB COLORS (*; }オブジェクト; 前景色; 背景色{; 代替背景色})

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>はフィールドまたは変数
オブジェクト	オブジェクト	→ オブジェクト名（“*”を指定した場合）、 またはフィールド または変数（“*”を省略した場合）
前景色	数値	→ 前景色のRGBカラーの値
背景色	数値	→ 背景色のRGBカラーの値
代替背景色	数値	→ 代替背景色のRGBカラーの値

4th Dimension 2004では、**SET RGB COLORS** コマンドに2つの新機能が追加されました。

■ 新しい任意の引数<代替背景色>

■ 新しい定数「automatic color」

代替背景色

SET RGB COLORS コマンドは、任意の引数<代替背景色>を新たに受け入れます。この引数は、設定されるオブジェクトがリストボックスまたはリストボックスの列である場合にのみ使用できます（リストボックスに関する詳細は、後述の「リストボックス」の節を参照）。

<代替背景色>により、偶数行の背景の代替色を設定することができます。この引数を使用すると、<代替背景色>は奇数行の背景にのみ使用されます。

代替色を使用すると、配列が見やすくなります。

<オブジェクト>により、リストボックスオブジェクトが指定された場合、代替色はリストボックス全体に使用されます。<オブジェクト>により、ある列が指定された場合、そのカラムにのみ指定した色が使用されます。

「Color」定数

SET RGB COLORS コマンドは、デフォルトとして<前景色>、<背景色>、<代替背景色>に新しいシステムカラー定数を受け入れます。明確を期するため、「**SET RGB COLOR**」の最初の4つの定数名が変更されています。

定数（以前の名前）	タイプ	値
Foreground color(Default foreground color)	倍長整数	-1
Background color(Default background color)	倍長整数	-2
Dark shadow color(Default dark shadow color)	倍長整数	-3
Light shadow color(Default light shadow color)	倍長整数	-4
Highlight text background color	倍長整数	-7
Highlight text color	倍長整数	-8
Highlight menu background color	倍長整数	-9
Highlight menu text color	倍長整数	-10
Disable highlight item color	倍長整数	-11

SET COLOR

SET COLOR ({*; }オブジェクト; カラー{; 代替色})

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>はフィールドまたは変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、またはフィールド または変数（“*”を省略した場合）
カラー	整数	→ オブジェクトの新しいカラー
代替色	整数	→ リストボックスの代替色

SET COLOR コマンドは、任意の引数<代替色>を新たに受け入れます。この引数は、指定されたオブジェクトがリストボックスタイプまたはリストボックスの列である場合にのみ使用できます。

<代替色>により、偶数行の代替色を設定することができます。この引数を渡すと、<カラー>は奇数行にのみ適用されます。代替色を使用すると、リストが読みやすくなります。

<オブジェクト>により、リストボックスオブジェクトが指定された場合、代替色はリストボックス全体に使用されます。<オブジェクト>により、ある列が指定された場合、そのカラムにのみ指定した色が使用されます。

リストボックス

この新しいテーマのコマンドは、新しいフォームオブジェクトタイプであるリストボックスを扱うために設けられました。

リストボックスは、以前のバージョンの4th Dimensionにおける“グループ化したスクロールエリア”に相当します。リストボックスはグループ化したスクロールエリアのすべての機能、特に列や選択可能な行の形式でデータを表示する機能を提供します。しかし、4th Dimension 2004のリストボックスには、値の入力、列の並び替え、代替色の定義など、さらに数多くの機能が用意されています。4th Dimensionの「フォーム」エディタでリストボックスオブジェクトタイプを完全に設定することが可能で、また、プログラムから管理することもできます。

この節では、プログラムを用いたリストボックスオブジェクトタイプの管理に関連する新機能について説明します。リストボックスタイプのオブジェクトの作成および設定に関する詳細は、前述の「リストボックス」の節を参照してください。

この節では次の事柄について説明します。

■ リストボックスオブジェクトタイプのプログラミング入門

■ リストボックスオブジェクトを管理するための新規コマンド：**INSERT LISTBOX COLUMN**、**DELETE LISTBOX COLUMN**、**Get number of listbox columns**、**SORT LISTBOX COLUMNS**、**SET LISTBOX COLUMN WIDTH**、**Get listbox column width**、**SELECT LISTBOX ROW**、**INSERT LISTBOX ROW**、**DELETE LISTBOX ROW**、**Get number of listbox rows**、**SET LISTBOX ROWS HEIGHT**、**Get listbox rows height**、**MOVED LISTBOX ROW NUMBER**、**GET LISTBOX ARRAYS**、**Get listbox information**、**SHOW LISTBOX GRID**、**SET LISTBOX GRID COLOR**、**SHOW LISTBOX SCROLLBAR**、**MOVED LISTBOX COLUMN NUMBER**

また、リストボックスタイプのオブジェクトの導入により、**SET COLOR**と**SET RGB COLORS** コマンドが変更されています。この変更については、前述の「オブジェクトプロパティ」の節で説明しています。

“リストボックス”オブジェクトタイプのプログラミング入門

“リストボックス”タイプのオブジェクトのプログラミングは、4th Dimensionの他のリストフォームオブジェクトと同じ方法で行われます。ただし、次の節で説明するように、特定のルールに従わなくてはなりません。

作成および値の管理

“リストボックス”オブジェクトには、1つ以上の列を納めることができます。それぞれの列は、一次元の4th Dimension配列に関連付けなくてはなりません。すべての配列タイプが使用可能ですが、ポインタ配列だけは利用できません。各列の表示フォーマットは、「フォーム」エディタや**SET FORMAT** コマンドを使用して指定することができます。

ランゲージを使用する場合、ハイレベル“リストボックス”コマンド (**INSERT LISTBOX ROW** など) ならびに配列操作コマンドを用いて、列の値 (データ入力と表示) を管理します。例えば、列の内容を初期化するには、次の命令を使用することができます。

ARRAY TEXT(カラム名; サイズ)

また、リストを使用することもできます。

LIST TO ARRAY(リスト名; カラム名)

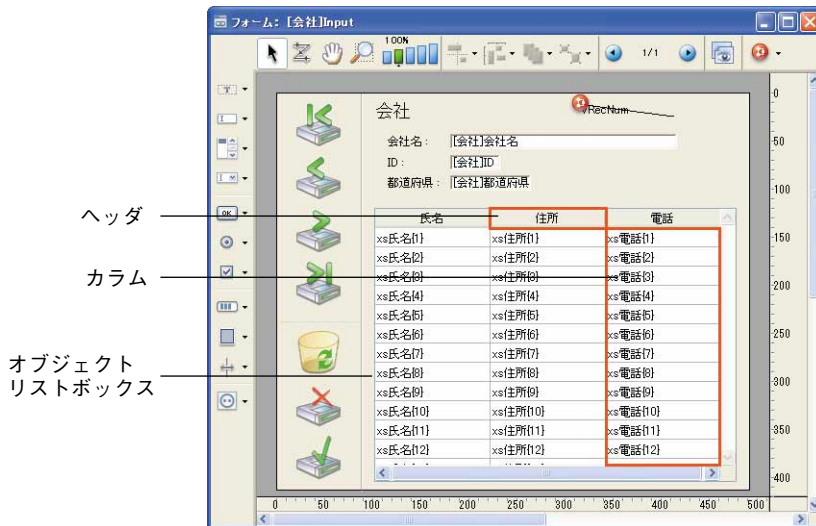
注：

- ・ リストボックスオブジェクトに複数の列が含まれている場合、関連する各配列のサイズは他の配列と同じでなければなりません（項目数が同じ）。同じサイズではない場合、最も小さな配列の項目数と同じ数だけ表示されます。
- ・ 4th Dimension 2004 では、2つの新規コマンド、Count in array と APPEND TO ARRAY が「配列」テーマに追加されました。

オブジェクト、カラム、ヘッダ

リストボックスオブジェクトは、オブジェクト自体、列、列ヘッダという異なる3つの項目で構成されます。これらの項目は「フォーム」エディタ上では別々に選択されます。

それぞれが独自のオブジェクト名や変数名を持ち、個別に処理されます。



デフォルトでは、リストボックスオブジェクトそのものとは関係なく、フォーム上の列にはカラム1からXまでの名前が付けられ、ヘッダにはヘッダ1からXまでの名前が付けられます。さらに注目すべき点は、オブジェクトや関連する変数にはデフォルトとして同じ名前が使用されることです。

各項目タイプには、独自の特性ならびに他の項目と共有する特性があります。例えば、文字のフォントはリストボックスオブジェクトに一括して割り当てることも、列やヘッダに対して個別に割り当てることもできます。これとは逆に、入力プロパティは列に対してのみ指定することができます。

このルールは、リストボックスと一緒に使用される「オブジェクトプロパティ」テーマのコマンドに対して適用されます。その機能に応じて、各コマンドをリストボックスや列、列ヘッダに対して使用します。作業を行おうとする項目のタイプを設定するには、その項目に関連付けた名前や変数を渡します。

次の表は、「オブジェクトプロパティ」テーマの各コマンドで使用可能なリストボックスオブジェクトタイプについて詳述したものです。

オブジェクトプロパティコマンド	オブジェクト	列	列ヘッダ
MOVE OBJECT	○		
GET OBJECT RECT	○		
SET FILTER		○	
SET FORMAT		○	
SET ENTERABLE		○	
SET CHOICE LIST		○	
BUTTON TEXT			○
SET COLOR	○	○	
SET RGB COLORS	○	○	
オブジェクトプロパティコマンド	オブジェクト	列	列ヘッダ
FONT	○	○	○
FONT SIZE	○	○	○
FONT STYLE	○	○	○
SET ALIGNMENT	○	○	○
Get alignment	○	○	○
SET VISIBLE	○	○	○
BEST OBJECT SIZE	○	○	○

注：SET LISTBOX COLUMN WIDTH（オブジェクト、列、ヘッダに適用される）と Get listbox column width（列とヘッダにのみ適用される）を除き、新しい「リストボックス」テーマのすべてのコマンドは、リストボックスオブジェクトに対してのみ適用されます。

オブジェクトメソッド

リストボックスオブジェクトやリストボックスの各列に対し、オブジェクトメソッドを付加することができます。オブジェクトメソッドの呼び出しは、次の順で行われます。

- 1.それぞれの列のオブジェクトメソッド
- 2.リストボックスのオブジェクトメソッド

列のオブジェクトメソッドにより、そのヘッダ上で発生したイベントを取得できます。

SET VISIBLE およびヘッダ

ヘッダで**SET VISIBLE** コマンドを使用すると、このコマンドで指定されたヘッダの枠を超えて、コマンドはすべてのリストボックスオブジェクトのヘッダに使用されます。例えば、**SET VISIBLE**(*;"ヘッダ3";False)という命令の場合、指定したヘッダだけではなく、ヘッダ3が属すリストボックスの全ヘッダを非表示にします。

Self

Self 関数（「ランゲージ」テーマ）は、リストボックスやリストボックスの列のオブジェクトメソッド内で使用することができます。双方の場合ともに、クリックされた場所に応じて、列の変数やヘッダの変数へのポインタが返されます。

Focus object

注：Focus object 関数は、既存のLast object 関数の新しい名前です（「ユーザインタフェース」テーマ）。明確を期するため、このコマンドの名前が変更されました。

Focus object 関数は、フォーカスを取得したリストボックスの列（つまり、配列）へのポインタを返します。4th Dimension のポインタのメカニズムを利用し、修正された配列の項目番号を調べることができます。例えば、ユーザが5行目の列“col2”を変更した場合は、次のようになります。

```
$Column:=Focus object
`$Columnには、col2へのポインタが代入される
$Row:= $Column->`$Rowは 5
```

EDIT ITEM

新しい**EDIT ITEM** コマンド（「入力制御」テーマ）を使用して、リストボックスオブジェクトのセルを編集モードに移行することができます。この件に関する詳細は、前述の**EDIT ITEM** コマンドを参照してください。

フォームイベント

プログラムによるリストボックス管理を円滑にするため、新しいフォームイベントが作成されました。詳細については、「リストボックスに関連するフォームイベント」の節を参照してください。

ヘッダ変数と並び替えの管理

デフォルトとして、ヘッダがクリックされると、リストボックスは自動的に標準的なコラムの並び替えを行います。標準的な並び替えとは、列の値をアルファベット順に並べ替え、続けてクリックされると昇順/降順を交互に切り替えます。常にすべての列は自動的に同期化されます。

リストボックスの「並び替え可」プロパティの選択を解除すると、ユーザによる標準の並び替えを禁止することができます。

開発者は、**SORT LISTBOX COLUMNS** コマンドを使用するか、または「On header click」と「On after sort」フォームイベント、および4Dの配列管理コマンドを組み合わせることにより、独自の並び替えを設定することができます（前述の「リストボックスに関連するフォームイベント」の節を参照）。

注：列のプロパティ「並び替え可」は、ユーザによる標準の並び替えにのみ影響を与えます。SORT LISTBOX COLUMNS コマンドでは、このプロパティが考慮されません。

列ヘッダに関連付けられた変数の値を使用すると、列の現在の並び替え状況（の読み込み）や並び替え矢印の表示など、追加情報を管理することができます。

■ 変数を0に設定すると、列は並び替えられず、並び替え矢印も表示されません。

氏名

■ 変数を1に設定すると、列は昇順に並び替えられ、並び替え矢印が表示されます。

氏名 ▲

■ 変数を2に設定すると、列は降順に並び替えられ、並び替え矢印が表示されます。

氏名 ▼

並び替え矢印を“強制的に”表示するために変数の値を設定することができます（例えば、ヘッダ2:=2）。この場合、列の並び替えそのものは変更されず、その処理は開発者に委ねられます。

オブジェクト変数と選択項目の管理

リストボックスオブジェクトに関連付けられた変数を使用して、オブジェクト行の選択項目の取得、設定、保存を行います。

この変数は、4th Dimensionが自動的に作成、保守を行うブールタイプの配列に対応します。また、この配列のサイズは、リストボックスのサイズにより決定されます。配列には列に関連付けられた最も小さな配列と同じ数の要素が格納されます。

この配列の各要素には、対応する行が選択された場合には“True”が、それ以外の場合は“False”が納められます。4th Dimensionは、ユーザの動作に応じて、この配列の内容を更新します。これとは逆に、配列要素の値を変更して、リストボックス中の選択項目を変えることができます。

その反面、この配列への行の挿入や削除はできず、行のタイプ変更もできません。

注：新しいCount in array関数は、選択された行の数を調べる際に役立ちます。

▼このメソッドにより、リストボックスの1行めの選択を逆転させます。

```
、 tListBox はリストボックス変数の名前です
If(tListBox{1} = True)
    tListBox{1}:= False
Else
    tListBox{1}:= True
End if
```

新しいコマンド

INSERT LISTBOX COLUMN

INSERT LISTBOX COLUMN ({*; }オブジェクト; 列位置; 列名; 列変数; ヘッダ名; ヘッダ変数)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
列位置	数値	→ 挿入する列の位置
列名	文字列	→ 列のオブジェクトの名前
列変数	配列	→ 列の変数の名前
ヘッダ名	文字列	→ 列ヘッダオブジェクトの名前
ヘッダ変数	整数変数	→ 列ヘッダの変数

INSERT LISTBOX COLUMN コマンドは、引数<オブジェクト>および<*>で指定されたリストボックスに列を挿入します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

新しい列は、引数<列位置>を用いて指定された列の直前に挿入されます。引数<列位置>の値が列の合計数よりも大きい場合、最後の列の後ろにカラムが追加されます。

引数<列名>と<列変数>には、挿入される列のオブジェクト名および変数名を渡します。この変数名は、列に表示する内容が格納された配列の名前と一致していません。

引数<ヘッダ名>と<ヘッダ変数>には、挿入される列のヘッダのオブジェクト名および変数を渡します。

注：オブジェクト名は、フォーム内で重複してはいけません。引数<列名>や<ヘッダ名>に渡される名前が既に使用されていないことを必ず確認してください。

重複した名前を指定すると、列は作成されず、エラーが生成されます。

▼ リストボックスの最後に列を追加しましょう。

```

C_LONGINT(HeaderVarName;$Last;$RecNum)
ALL RECORDS([Table1])
$RecNum:=Records in table([Table1])
ARRAY PICTURE(Picture;$RecNum)
$Last:=Get number of listbox columns(";listbox1")+1
INSERT LISTBOX COLUMN(";listbox1";$Last;"ColumnPicture";Picture;
"HeaderPicture";HeaderVarName)

```

DELETE LISTBOX COLUMN

DELETE LISTBOX COLUMN ({*; }オブジェクト; 位置{; 列数})

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
列位置	数値	→ 削除する列の番号
列数	数値	→ 削除する列の数

DELETE LISTBOX COLUMN コマンドは、引数<オブジェクト>および<*>で指定されたリストボックスから1つ以上の列（表示または非表示の列）を削除します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

任意の引数である<列数>を渡さない場合、コマンドは引数<列位置>で指定された列だけを削除します。

この引数を渡した場合、引数<列数>は、引数<列位置>より（この列を含む）右側にある削除する列の数を示します。

引数<列位置>がリストボックスの列の数よりも大きい場合、コマンドは何も行いません。

Get number of listbox columns

Get number of listbox columns ({*; }オブジェクト) → 倍長整数

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
戻り値	倍長整数	← 列の数

Get number of listbox columns 関数は、引数<オブジェクト>および<*>で指定されたリストボックスに存在する列（表示または非表示の列）の合計数を返します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

SORT LISTBOX COLUMNS

SORT LISTBOX COLUMNS ({*; }オブジェクト; 列変数; 順序{; 列変数2; 順序2})...{; 列変数X; 順序X)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、または変数（“*”を省略した場合）
列変数1...X	変数	→ 並び替える列の変数
順序1...X	> または <	→ >：昇順に並び替え、または <：降順に並び替え

SORT LISTBOX COLUMNS コマンドは、1つ以上の列の値に基づいて、引数<オブジェクト>および<*>で指定されたリストボックスの行を並べ替えます。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

<列変数>には、列に関連付けられた変数（つまり、配列名）を渡し、この列の値が並び替え条件として使用されます。ピクチャタイプとポインタタイプを除き、任意のタイプの配列データを使用することができます。

<順序>には、並び替え順を表わす“>”または“<”記号を渡します。<順序>に“～より大きい”を示す記号(>)を指定すると、並び替えは昇順になります。<順序>に“～より小さい”を示す記号(<)を指定すると、並び替えは降順になります。

さらに、マルチレベルソートを定義することができます。これを行うには、必要な数のペア（列変数;順序）を渡します。並び替えレベルは、コマンド内の引数の位置によって決まります。

リストボックス操作の原則に従い、それぞれの列は同期化されます。つまり、列の並び順は、自動的にそのオブジェクトの他のすべての列に受け継がれます。

SET LISTBOX COLUMN WIDTH

SET LISTBOX COLUMN WIDTH ({*; }オブジェクト; 幅)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
幅	整数	→ 列の幅（ピクセル単位）

SET LISTBOX COLUMN WIDTH コマンドを使用し、引数<オブジェクト>および<*>で指定されたオブジェクト（リストボックス、列、またはヘッダ）の任意の列の幅、またはすべての列の幅をプログラムから変更することができます。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

引数<幅>には、<オブジェクト>の新しい幅（ピクセル単位）を渡します。

- 指定された<オブジェクト>がリストボックスオブジェクトである場合、リストボックスのすべての列サイズが変更されます。
- 指定された<オブジェクト>が列または列ヘッダである場合、その列のサイズだけが変更されます。

Get listbox column width

Get listbox column width ({*; }オブジェクト) → 整数

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
戻り値	整数	← 列の幅（ピクセル単位）

Get listbox column width 関数は、引数<オブジェクト>および<*>で指定された列の幅（ピクセル単位）を返します。

引数<オブジェクト>には、リストボックスの列や列ヘッダを自由に渡すことができます。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

MOVED LISTBOX COLUMN NUMBER

MOVED LISTBOX COLUMN NUMBER ({*;}オブジェクト; 旧位置; 新位置)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、または変数（“*”を省略した場合）
旧位置	数値	← 移動された列の以前の位置
新位置	数値	← 移動された列の新しい位置

MOVED LISTBOX COLUMN NUMBER コマンドは、<旧位置>と<新位置>に2つの数値を返し、引数<オブジェクト>および<*>で指定されたリストボックス内で移動された列の以前の位置と新しい位置をそれぞれ示します。このコマンドは、「**On column moved**」フォームイベントと一緒に使用しなくてはなりません（前述の「リストボックスに関連するフォームイベント」の節を参照）。

注：このコマンドは、非表示の列を考慮します。

SELECT LISTBOX ROW

SELECT LISTBOX ROW ({*;}オブジェクト; 位置{; アクション})

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、または変数（“*”を省略した場合）
位置	倍長整数	→ 選択する行の番号
アクション	倍長整数	→ 選択に関する動作

SELECT LISTBOX ROW コマンドは、引数<オブジェクト>および<*>で指定されたリストボックス内において、<位置>に渡した番号の行を選択します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

任意の引数<アクション>を指定すると、その値を用いて、行のセレクションがリストボックス内に既に存在している場合に実行する選択アクションが決定されます。この引数には、任意の値または次の定数のいずれかを渡すことができます（「リストボックス」テーマ）。

■ **Replace listbox selection(0)**：選択した行は新しいセレクションとなり、既存のものと同じ置き換えられます。このコマンドは、ユーザが行をクリックした場合と同じ結果になります。

デフォルトでは、この動作が実行されます（引数<アクション>を省略した場合）。

■ **Add to listbox selection(1)**：選択した行は既存のセレクションに追加されます。指定した行が既存のセレクションに属している場合、コマンドは何も行いません。

■ **Remove from listbox selection(2)**：選択した行は既存のセレクションから削除されます。指定した行が既存のセレクションに属さない場合、コマンドは何も行いません。

INSERT LISTBOX ROW

INSERT LISTBOX ROW ({*; }オブジェクト; 位置)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
位置	倍長整数	→ 行を挿入する位置

INSERT LISTBOX ROW コマンドは、引数<オブジェクト>および<*>で指定されたリストボックスに新しい行を挿入します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

行は、引数<位置>で指定された場所に挿入されます。配列のタイプや表示状態とは関係なく、リストボックスの列に使用されるすべての配列に対し、この場所に新しい行が自動的に追加されます。

<位置>の値がリストボックス内の合計行数よりも大きい場合、行は各配列の最後に追加されます。

DELETE LISTBOX ROW

DELETE LISTBOX ROW ({*; }オブジェクト; 位置)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
位置	倍長整数	→ 削除する行の位置

DELETE LISTBOX ROW コマンドは、引数<オブジェクト>および<*>で指定されたリストボックス（表示または非表示）から、<位置>に指定した番号の行を削除します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

<位置>で指定した行は、リストボックスの列で使用されるすべての配列から自動的に削除されます。

<位置>の値がリストボックス内の合計行数よりも大きい場合、コマンドは何も行いません。

Get number of listbox rows

Get number of listbox rows ({*; }オブジェクト) → 倍長整数

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名 (文字列) 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名 (“*” を指定した場合)、 または変数 (“*” を省略した場合)
戻り値	倍長整数	← 行数

Get number of listbox rows 関数は、引数<オブジェクト>および<*>で指定されたリストボックスの行の数を返します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名 (文字列)であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

SET LISTBOX ROWS HEIGHT

SET LISTBOX ROWS HEIGHT{*; }オブジェクト; 高さ)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名 (文字列) 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名 (“*” を指定した場合)、 または変数 (“*” を省略した場合)
高さ	整数	→ 行の高さ (ピクセル単位)

SET LISTBOX ROWS HEIGHT コマンドを使用すると、引数<オブジェクト>および<*>で指定されたリストボックス内の行の高さをプログラムにより変更することができます。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名 (文字列)であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

Get listbox rows height

Get listbox rows height ({*;}オブジェクト) → 整数

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
戻り値	整数	← 行の高さ（ピクセル単位）

Get listbox rows height関数は、引数<オブジェクト>および<*>で指定されたリストボックスのカレント行の高さ（ピクセル単位）を返します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

MOVED LISTBOX ROW NUMBER

MOVED LISTBOX ROW NUMBER ({*;}オブジェクト; 旧位置; 新位置)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
旧位置	数値	← 移動された行の以前の位置
新位置	数値	← 移動された行の新しい位置

MOVED LISTBOX ROW NUMBER コマンドは、<旧位置>と<新位置>に2つの数値を返し、引数<オブジェクト>および<*>で指定されたリストボックス内で移動された行の以前の位置と新しい位置をそれぞれ示します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

このコマンドは、「**On row moved**」フォームイベントと一緒に使用しなくてはなりません（前述の「リストボックスに関連するフォームイベント」の節を参照）。

GET LISTBOX ARRAYS

GET LISTBOX ARRAYS (*; }オブジェクト; 列名配列; ヘッダ名配列; 列変数配列; ヘッダ変数配列; 表示配列; スタイル配列)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、または変数（“*”を省略した場合）
列名配列	文字列配列	← 列のオブジェクトの名前
ヘッダ名配列	文字列配列	← ヘッダオブジェクトの名前
列変数配列	ポインタ配列	← 列の変数へのポインタ
ヘッダ変数配列	ポインタ配列	← ヘッダの変数へのポインタ
表示配列	ブール配列	← 各列の表示状態
スタイル配列	ポインタ配列	← スタイルおよびカラー配列へのポインタ

GET LISTBOX ARRAYS コマンドは一連の同期化された配列を返し、引数<オブジェクト>および<*>で指定されたリストボックスの各列（表示または非表示の列）に関する情報を提供します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

コマンドの実行後、各引数には次の値が代入されます。

- 配列<列名配列>には、リストボックス内の各列のオブジェクト名一覧が代入されます。
- 配列<ヘッダ名配列>には、リストボックス内の各列ヘッダのオブジェクト名一覧が代入されます。
- 配列<列変数配列>には、リストボックス内の各列に関連付けられた変数（配列）へのポインタが代入されます。
- 配列<ヘッダ変数配列>には、リストボックス内の各列ヘッダに関連付けられた変数へのポインタが代入されます。
- 配列<表示配列>には各列に関するブール値が代入され、リストボックス内の列が表示（True）または非表示（False）のいずれであるかを示します。

■ 配列<スタイル配列>には、3つの配列をそれぞれ指す3つのポインタが代入されます。これら3つの配列は、リストボックス内の各行に対して特定のスタイルやフォントカラー、背景色を適用するために使用されます。

これらの配列は、「デザイン」モードの「プロパティリスト」上の「リストボックス」テーマに関連しています（詳細については、前述の「リストボックス固有のプロパティ」の節を参照）。

リストボックスに対する配列が指定されていない場合、<スタイル配列>内の対応する項目にはNilポインタが代入されます。

Get listbox information

Get listbox information ({*; }オブジェクト; 情報) → 倍長整数

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、または変数（“*”を省略した場合）
情報	倍長整数	→ 取得する情報
戻り値	倍長整数	← 現在の値

Get listbox information 関数は、引数<オブジェクト>および<*>で指定されたリストボックスオブジェクトの各種情報を返します。これは、現在の表示状態やヘッダサイズ、スクロールバーに関する情報です。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

引数<情報>には、取得しようとする情報のタイプを示す値を渡します。この引数には任意の値または「リストボックス」テーマの次の定数のいずれかを使用することができます。

定数	タイプ	値	戻り値
Display listbox header	倍長整数	0	0=非表示、1=表示
Listbox header height	倍長整数	1	高さ (ピクセル単位)
Display listbox hor scrollbar	倍長整数	2	0=非表示、1=表示
Listbox hor scrollbar height	倍長整数	3	高さ (ピクセル単位)
Display listbox ver scrollbar	倍長整数	4	0=非表示、1=表示
Listbox ver scrollbar width	倍長整数	5	幅 (ピクセル単位)

このコマンドは、フォーム上のリストボックスエリアの実サイズを計算する際に役立ちます。

SHOW LISTBOX GRID

SHOW LISTBOX GRID ({*; }オブジェクト; 横線; 縦線)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名 (文字列) 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名 (“*” を指定した場合)、または変数 (“*” を省略した場合)
横線	ブール	→ True=表示、False=非表示
縦線	ブール	→ True=表示、False=非表示

SHOW LISTBOX GRID コマンドを使用し、水平や垂直グリッドラインを表示、または非表示に設定することができます。これらのグリッドラインは、引数<オブジェクト>および<*>で指定されたリストボックスオブジェクト上のグリッドを構成します。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名 (文字列) であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

<横線>と<縦線>には、対応するグリッドラインを表示するか (True)、表示しないか (False) を示すブール値を渡します。デフォルトでは、グリッドが表示されます。

参照

SET LISTBOX GRID COLOR

SET LISTBOX GRID COLOR

SET LISTBOX GRID COLOR{*; }オブジェクト; カラー; 横線; 縦線)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、または変数（“*”を省略した場合）
カラー	数値	→ RGBカラーの値
横線	ブール	→ 水平グリッドラインに使用されるカラー
縦線	ブール	→ 垂直グリッドラインに使用されるカラー

SET LISTBOX GRID COLOR コマンドを使用し、引数<オブジェクト>および<*>で指定されたリストボックスオブジェクト上のグリッドの色を変更することができます。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

<カラー>には、RGBカラーの値を渡します。RGBカラーに関する詳細は、**SET RGB COLORS** コマンドの説明を参照してください。

<横線>と<縦線>を使用し、カラーを適用するグリッドラインを指定することができます。

■ <横線>に“True”を渡した場合、水平グリッドラインにカラーが適用されます。“False”を渡すと、カラーは変更されません。

■ <縦線>に“True”を渡した場合、垂直グリッドラインにカラーが適用されます。“False”を渡すと、カラーは変更されません。

参照

SHOW LISTBOX GRID

SHOW LISTBOX SCROLLBAR

SHOW LISTBOX SCROLLBAR({*; }オブジェクト; 横; 縦)

引数	タイプ	説明
*		→ 指定した場合、<オブジェクト>はオブジェクト名（文字列） 省略した場合、<オブジェクト>は変数
オブジェクト	フォームオブジェクト	→ オブジェクト名（“*”を指定した場合）、 または変数（“*”を省略した場合）
横	ブール	→ True=表示、False=非表示
縦	ブール	→ True=表示、False=非表示

SHOW LISTBOX SCROLLBAR コマンドを使用すると、引数<オブジェクト>および<*>で指定されたリストボックスオブジェクトにおいて、横スクロールバーや縦スクロールバーを表示、または非表示に設定することができます。

任意の引数<*>を渡すことにより、引数<オブジェクト>がオブジェクト名（文字列）であることを示します。この引数を渡さない場合、引数<オブジェクト>が変数であることを示します。その場合、文字列ではなく、変数の参照番号を渡してください。

<横>と<縦>には、対応するスクロールバーを表示するか（True）、表示しないか（False）を示すブール値を渡します。デフォルトでは、スクロールバーが表示されます。

4 D 環境

OPEN 4D PREFERENCES

OPEN 4D PREFERENCES(セレクト)

引数	タイプ	説明
セレクト	文字列	→ 「環境設定」ダイアログボックスの テーマ、ページ、パラメータグループ を指定するキー

OPEN 4D PREFERENCES コマンドは、現行の4th Dimensionアプリケーションの「環境設定」ダイアログボックスを表示し、<セレクト>に渡されたキーに対応するテーマやページを表示します。

引数<セクタ>には、「環境設定」ダイアログボックスのテーマやページ、パラメータグループを示す“キー”を1つ以上納めなくてはなりません。これらキーの一覧は、この節の最後に記載されています。

固定アクセスパスまたは単独項目の名前のいずれかを渡すことができます。

■ **固定アクセスパス**：引数<セクタ>は次の方法で組み立てることができます：/テーマ{/ページ{/パラメータグループ}}

文字列は必ず“/”記号で開始し、各レベルを“/”で区切らなくてはなりません。

例えば、「デザインモード」ページの「コンパイラ」ページを設定する場合、引数<セクタ>には“/Design Mode/Compiler”を代入します。

■ **名前（相対パス）**：この場合、引数<セクタ>は“/”記号で開始しません。目的の項目の名前を渡すだけで、4th Dimensionは「パラメータグループ→ ページ→ テーマ」という検索順に従い、最初に一致する項目を開きます。

例えば、<セクタ>に“Progress indicator”を渡した場合、4th Dimensionは「アプリケーション」テーマの「オプション」ページを開きます。

ダイアログボックスの最初のページを直接開くには、<セクタ>に“/”を渡します。

コマンドは、<セクタ>に指定された項目に関する「環境設定」ページを開きますが、その他すべてのテーマやページにもアクセスすることができます。ユーザが「環境設定」にアクセスすることにより、アプリケーションの妨げとにならないよう確認することは開発者の責任です。

ユーザ動作を制御するために、ユーザアクセス管理システムを有効にすることをお勧めします。

要求した項目が見つからない場合や、正常に開けなかった場合は、システム変数OKは1を返し、それ以外の場合は0を返します。

▼「環境設定」の最初のページを開く

OPEN 4D PREFERENCES("/")

▼「アプリケーション」テーマの「ショートカット」ページを開く

OPEN 4D PREFERENCES("/Application/Shortcuts")

▼「デザインモード」テーマの「メソッドエディタ」ページを開く

OPEN 4D PREFERENCES("Default Display")

パスのキー

引数<セクタ>として使用できるキーの一覧を次に示します。

/Application（アプリケーション）

/Application/Options (アプリケーション/オプション)
/Application/Options/Options (アプリケーション/オプション/オプション)
/Application/Options/Temporary Folder Location (アプリケーション/オプション/テンポラリフォルダの場所)
/Application/Options/Drag and Drop Highlight (アプリケーション/オプション/ドラッグドロップ点滅)
/Application/Options/Progress Indicator (アプリケーション/オプション/進捗インジケータ)
/Application/Options/Display Toolbar (アプリケーション/オプション/ツールバー表示)
/Application/Options/Display Windows (アプリケーション/オプション/ウインドウを表示する)
/Application/Access (アプリケーション/アクセス管理)
/Application/Access/Data Access (アプリケーション/アクセス管理/データアクセス権)
/Application/Access/User Access (アプリケーション/アクセス管理/ユーザアクセス)
/Application/CPU Priorities (アプリケーション/CPUの優先度)
/Application/CPU Priorities/Set CPU Priority to: (アプリケーション/CPUの優先度/他のアプリケーション)
/Application/Shortcuts (アプリケーション/ショートカット)
/Application/Shortcuts/Keys (アプリケーション/ショートカット/編集)
/Application/Compatibility (アプリケーション/互換)
/Application/Compatibility/Structure Compatibility (アプリケーション/互換/ストラクチャの互換性)
/Application/Compatibility/Web Compatibility (アプリケーション/互換/Web互換性)
/Application/Compatibility/Platform (アプリケーション/互換/プラットフォーム)
/Design Mode (デザインモード)
/Design Mode/Structure (デザインモード/ストラクチャ)
/Design Mode/Structure/General Font (デザインモード/ストラクチャ/通常フォント)
/Design Mode/Structure/Forms and Methods Automatic Comments (デザインモード/ストラクチャ/フォーム/メソッド用自動コメント)
/Design Mode/Form Editor (デザインモード/フォームエディタ)
/Design Mode/Form Editor/Object Templates (デザインモード/フォームエディタ/オブジェクトテンプレート)
/Design Mode/Form Editor/Move (デザインモード/フォームエディタ/移動)
/Design Mode/Form Editor/Auto Alignment (デザインモード/フォームエディタ/自動整列)
/Design Mode/Form Editor/Default Display (デザインモード/フォームエディタ/デフォルト表示)
/Design Mode/Method Editor (デザインモード/メソッドエディタ)
/Design Mode/Method Editor/Font (デザインモード/メソッドエディタ/フォント)
/Design Mode/Method Editor/Default Display (デザインモード/メソッドエディタ/デフォルト表示)
/Design Mode/Method Editor/Options (デザインモード/メソッドエディタ/オプション)

/Design Mode/Method Editor/Structure Style-Sheets (デザインモード/メソッドエディタ/
ストラクチャスタイルシート)
/Design Mode/Compiler (デザインモード/コンパイラ)
/Design Mode/Compiler/Compilation Options (デザインモード/コンパイラ/コンパイルオ
プション)
/Design Mode/Compiler/Compiler Methods for... (デザインモード/コンパイラ/コンパイラ
メソッド...)
/Design Mode/Documentation (デザインモード/ドキュメント)
/Design Mode/Documentation/Documentation Access from the Explorer (デザインモード/
ドキュメント/エクスプローラからのドキュメントアクセス)
/Database (データベース)
/Database/Data Management (データベース/データ管理)
/Database/Data Management/General (データベース/データ管理/一般設定)
/Database/Data Management/Database Cache Settings (データベース/データ管理/デー
タベースキャッシュ設定)
/Database/Data Management/WEDD (データベース/データ管理/WEDD)
/Database/Script Manager (データベース/スクリプトマネージャ)
/Database/Script Manager/Script Manager (データベース/スクリプトマネージャ/スクリ
プトマネージャ)
/Backup (バックアップ)
/Backup/Configuration (バックアップ/設定)
/Backup/Configuration/Backup Contents (バックアップ/設定/バックアップの内容)
/Backup/Configuration/Backup File Destination Folder (バックアップ/設定/バックアップ
ファイル保存先フォルダ)
/Backup/Configuration/Last Backup Information (バックアップ/設定/最後のバックアップ
情報)
/Backup/Configuration/Log Management (バックアップ/設定/ログ管理)
/Backup/Scheduler (バックアップ/スケジューラ)
/Backup/Scheduler/Backup Frequency (バックアップ/スケジューラ/バックアップスケ
ジュール)
/Backup/Backup (バックアップ/バックアップ)
/Backup/Backup/General (バックアップ/バックアップ/一般設定)
/Backup/Backup/Archive (バックアップ/バックアップ/アーカイブ)
/Backup/Restore (バックアップ/復元)
/Backup/Restore/Automatic Restore (バックアップ/復元/自動復元)
/Client-Server (クライアント-サーバ)
/Client-Server/Configuration (クライアント-サーバ/設定)
/Client-Server/Configuration/Network (クライアント-サーバ/設定/ネットワーク)
/Client-Server/Configuration/Client-Server Connections Timeout (クライアント-サーバ/設
定/クライアント-サーバ接続タイムアウト)

/Client-Server/Configuration/Client-Server Communication (クライアント-サーバ/設定/クライアント-サーバ通信)
/Client-Server/Configuration/4D Open (クライアント-サーバ/設定/4D Open)
/Client-Server/Publishing (クライアント-サーバ/公開)
/Client-Server/Publishing/Publishing Information (クライアント-サーバ/公開/公開情報)
/Client-Server/Publishing/Allow-Deny Table Configuration (クライアント-サーバ/公開/許可-拒否テーブル設定)
/Client-Server/Publishing/Encryption (クライアント-サーバ/公開/暗号化)
/Web (Web)
/Web/Configuration (Web/設定)
/Web/Configuration/Web Server Publishing (Web/設定/Web サーバ公開時オプション)
/Web/Configuration/Default HTML Path (Web/設定/デフォルトHTMLパス)
/Web/Configuration/Starting Mode (Web/設定/開始時のモード)
/Web/Advanced (Web/詳細)
/Web/Advanced/Cache (Web/詳細/キャッシュ)
/Web/Advanced/Web Process (Web/詳細/Web プロセス)
/Web/Advanced/Options (Web/詳細/オプション)
/Web/Advanced/Web Passwords (Web/詳細/Web パスワード)
/Web/Options (Web/オプション)
/Web/Options/Text Conversion (Web/オプション/テキスト変換)
/Web/Options/4D WebSTAR (Web/オプション/4D WebSTAR)
/Web/Options/Persistent Connections (Web/オプション/keep Alive 接続)
/Web Services (Web サービス)
/Web Services/SOAP (Web サービス/SOAP)
/Web Services/SOAP/Server Side (Web サービス/SOAP/サーバ側設定)
/Web Services/SOAP/Client Side (Web サービス/SOAP/クライアント側設定)
/Application
/Application/Options
/Application/Options/Options
/Application/Options/Temporary Folder Location
/Application/Options/Drag and Drop Highlight
/Application/Options/Progress Indicator
/Application/Options/Display Toolbar
/Application/Options/Display Windows
/Application/Access
/Application/Access/Data Access
/Application/Access/User Access
/Application/CPU Priorities
/Application/CPU Priorities/Set CPU Priority to:
/Application/Shortcuts
/Application/Shortcuts/Keys

- /Application/Compatibility
- /Application/Compatibility/Structure Compatibility
- /Application/Compatibility/Web Compatibility
- /Application/Compatibility/Platform
- /Design Mode
- /Design Mode/Structure
- /Design Mode/Structure/General Font
- /Design Mode/Structure/Forms and Methods Automatic Comments
- /Design Mode/Structure/Structure Editor
- /Design Mode/Form Editor
- /Design Mode/Form Editor/Object Templates
- /Design Mode/Form Editor/Move
- /Design Mode/Form Editor/Auto Alignment
- /Design Mode/Form Editor/Default Display
- /Design Mode/Form Editor/Markers
- /Design Mode/Method Editor
- /Design Mode/Method Editor/Font
- /Design Mode/Method Editor/Default Display
- /Design Mode/Method Editor/Options
- /Design Mode/Method Editor/Styles for Syntax Elements
- /Design Mode/Compiler
- /Design Mode/Compiler/Compilation Options
- /Design Mode/Compiler/Compiler Methods for...
- /Design Mode/Documentation
- /Design Mode/Documentation/Documentation Access from the Explorer
- /Database
- /Database/Data Management
- /Database/Data Management/General Settings
- /Database/Data Management/Database Cache Settings
- /Database/Data Management/WEDD
- /Database/Script Manager
- /Database/Script Manager/Script Manager
- /Backup
- /Backup/Configuration
- /Backup/Configuration/Backup Contents
- /Backup/Configuration/Backup File Destination Folder
- /Backup/Configuration/Last Backup Information
- /Backup/Configuration/Log Management
- /Backup/Scheduler
- /Backup/Scheduler/Backup Frequency
- /Backup/Backup

/Backup/Backup/General Settings
/Backup/Backup/Archive
/Backup/Restore
/Backup/Restore/Automatic Restore
/Client-Server
/Client-Server/Configuration
/Client-Server/Configuration/Network
/Client-Server/Configuration/Client-Server Connections Timeout
/Client-Server/Configuration/Client-Server Communication
/Client-Server/Configuration/4D Open
/Client-Server/Publishing
/Client-Server/Publishing/Publishing Information
/Client-Server/Publishing/Allow-Deny Table Configuration
/Client-Server/Publishing/Encryption
/Web
/Web/Configuration
/Web/Configuration/Web Server Publishing
/Web/Configuration/Default HTML Path
/Web/Configuration/Starting Mode
/Web/Advanced
/Web/Advanced/Cache
/Web/Advanced/Web Process
/Web/Advanced/Options
/Web/Advanced/Web Passwords
/Web/Options
/Web/Options/Text Conversion
/Web/Options/4D WebSTAR
/Web/Options/Keep-Alive Connections
/Web Services
/Web Services/SOAP
/Web Services/SOAP/Server Side
/Web Services/SOAP/Client Side

システム環境

SET ENVIRONMENT VARIABLE

SET ENVIRONMENT VARIABLE(変数名; 変数値)

引数	タイプ	説明
変数名	文字列	→ 設定する変数名
変数値	文字列	→ 変数の値、または デフォルト値にリセットする場合は、""

このコマンドを使用し、Mac OS XやWindowsにおける環境変数の値を設定することができます。このコマンドは、**SET CGI EXECUTABLE** コマンドまたは **LAUNCH EXTERNAL PROCESS** コマンドとともに使用する必要があります。

<変数名>には定義する変数の名前を、<変数値>にはその値を渡します。

環境変数とその値の総一覧を取得する方法については、お使いのOSのテクニカルドキュメントを参照してください。**SET CGI EXECUTABLE** コマンドで使用できる環境変数の一覧を調べるには、CGIに関する4Dのドキュメントを参照してください。

LAUNCH EXTERNAL PROCESS コマンドで使用できる環境変数の一覧を調べるには、このコマンドに関する説明を参照してください。

LAUNCH EXTERNAL PROCESS

LAUNCH EXTERNAL PROCESS(ファイル名{; 入力ストリーム{; 出力ストリーム{; エラーストリーム}}})

引数	タイプ	説明
ファイル名	文字列	→ 起動するファイルのファイルパスと引数
入力ストリーム	文字列	→ 入力ストリーム (stdin)
出力ストリーム	文字列	← 出力ストリーム (stdout)
エラーストリーム	文字列	← エラーストリーム (stderr)

注：このコマンドは、上級ユーザを対象に作成されています。

LAUNCH EXTERNAL PROCESS コマンドを使用して、Mac OS XやWindows上の4th Dimensionから外部プロセスを起動することができます。

Mac OS Xにおいて、このコマンドは「ターミナル」から起動できるあらゆるアプリケーションへのアクセスを提供します。

注：4D Pack ユーザの場合、このコマンドは AP_Sublaunch コマンドと同じ機能を持ちます。

引数<ファイル名>には、実行するアプリケーションの固定ファイルパス、ならびに必要な引数（必要な場合）を渡します。

Mac OS X 上では、アプリケーション名だけを渡すこともできます。4th Dimension は PATH 環境変数を使用して、実行形式ファイルの場所を探し出します。

引数<入力ストリーム>（任意）には、外部プロセスの stdin を納めます。コマンドが実行されると、引数<出力ストリーム>と<エラーストリーム>（渡した場合）には、外部プロセスの stdout と stderr がそれぞれ返されます。

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、それ以外の場合は（ファイルが見つからない、メモリが不十分など）0 が代入されます。

Mac OS X の例題

次の例題は、「アプリケーション/ユーティリティ」フォルダにある Mac OS X ターミナルを使用します。

▼ ファイルに対するパーミッションを変更するには、次のように記述します（chmod は Mac OS X のコマンドであり、ファイルアクセスを変更する際に使用します）。

```
LAUNCH EXTERNAL PROCESS("chmod+x/folder/myfile.txt")
```

▼ テキストファイルを編集するには、次のように記述します（cat は Mac OS X のコマンドであり、ファイルの編集を行う）。この例では、コマンドへのフルアクセスパスを渡しています。

```
C_TEXT(input;output)
```

```
input :=""
```

```
LAUNCH EXTERNAL PROCESS("/bin/cat/folder/myfile.txt"; input;output)
```

▼ 「Users」フォルダの内容を取得するには、次のように記述します（ls-l は Mac OS X のコマンドであり、DOS の dir コマンドに相当）。

```
C_TEXT($in;$out)
```

```
LAUNCH EXTERNAL PROCESS("/bin/ls-l/Users";$in;$out)
```

Windows の例題

▼ ノートパッドを開くには、次のように記述します。

```
LAUNCH EXTERNAL PROCESS("C:¥¥WINDOWS¥¥notepad.exe")
```

- ▼ ノートパッドを開いて特定のドキュメントをオープンするには、次のように記述します。

```
LAUNCH EXTERNAL PROCESS("C:¥¥WINDOWS¥¥notepad.exe
C:¥¥Docs¥¥new folder¥¥res.txt")
```

- ▼ Microsoft®Word®アプリケーションを起動し、特定のドキュメントをオープンするには、次のように記述します（""が2回使用されている点に注意）。

```
$mydoc:="C:¥¥Program Files¥¥Microsoft Office¥¥Office10
¥¥WINWORD.EXE¥¥C:¥¥Documents and Settings¥¥Bob¥¥Desktop
¥¥My Documents¥¥New folder¥¥test.xml¥¥"
```

```
LAUNCH EXTERNAL PROCESS($mydoc;$tIn;$tOut)
```

- ▼ Perlスクリプトを実行するには、次のように記述します（ActivePerlが必要）。

```
C_TEXT($input;$output)
SET ENVIRONMENT VARIABLE("myvariable";"value")
LAUNCH EXTERNAL PROCESS("D:¥¥Perl¥¥bin¥¥perl.exe D:¥¥Perl¥¥eg
¥¥cgi¥¥env.pl";$input;$output)
```

セレクション

4th Dimension 2004において、「セレクション」テーマにはセレクションの管理をリストフォームやサブフォームで行うための拡張機能が導入されました。次の項目が変更されています。

- 新しいGET HIGHLIGHTED RECORDS コマンドおよびDisplayed line number 関数
- HIGHLIGHT RECORDS コマンドの変更
- DISPLAY SELECTION およびMODIFY SELECTION コマンドに関する変更

注：この他の新機能により、リストやサブフォームがさらに管理しやすくなりました。

- ・ SET ENTERABLE コマンド（「オブジェクトプロパティ」テーマ）を使用して、リストフォームを編集可に設定することができます。
- ・ リストやサブフォームにおいて、新しいフォームイベントである「On Selection Change」が生成されます（前述の「フォームイベント」の節を参照）。
- ・ 新しいEDIT ITEM コマンド（「オブジェクトプロパティ」テーマ）を使用して、サブフォーム内の行を編集可に設定することができます。

GET HIGHLIGHTED RECORDS

GET HIGHLIGHTED RECORDS({テーブル; }セット名)

引数	タイプ	説明
テーブル	テーブル	→ セット UserSet がコピーされるテーブル省略した場合、カレントフォームのテーブル
セット名	文字列	→ セット UserSet がコピーされるセット

GET HIGHLIGHTED RECORDS コマンドは、引数として渡した<テーブル>のカレントセット UserSet をコピーして、引数<セット名>で指定したセットに格納します。引数<テーブル>を省略すると、カレントフォームやサブフォームのテーブルが使用されます。

覚書：テーブルのシステムセット UserSet には、出力フォームにおいてユーザが“強調表示”したレコードがすべて納められます。テーブルおよびプロセスごとに UserSet は1つだけ存在します。

このコマンドにより、強調表示されたレコードを受け取るテーブルを選定することができるため、**GET HIGHLIGHTED RECORDS** コマンドはサブフォーム内のレコードセレクションの管理を行うこともできます。この場合、サブフォームのセレクションは別のテーブルのものでも構いません。以前のバージョンの 4th Dimension では、ユーザ定義のセレクションの管理は UserSet を用いてのみ行われ、従ってカレントテーブルに限定されていました。

また、**GET HIGHLIGHTED RECORDS** コマンドはフォーム以外のコンテキストでも呼び出せますが、その場合は空のセットが返されます。

<セット名>で指定されたセットは、ローカルプロセスやクライアントプロセス、またはインタープロセスのいずれでも構いません。

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、それ以外の場合は0が代入されます。

▼ 次のメソッドは、テーブル ([CDs]) のレコードを表示するサブフォームにおいて、選択されているレコードの数を示します。

```
GET HIGHLIGHTED RECORDS([CDs];"$highlight")
ALERT(String(Records in set("$highlight"))+"レコードが選択されています。")
CLEAR SET("$highlight")
```

参照

HIGHLIGHT RECORDS

Displayed line number

Displayed line number → 倍長整数

引数	タイプ	説明
----	-----	----

このコマンドには引数は必要ありません。

戻り値	倍長整数	← 表示されている行の数
-----	------	--------------

Displayed line number 関数は「On display detail」フォームイベントでのみ機能します。関数は、画面上にレコードリストが表示される間に処理される行の数を返します。

表示された行が空ではない場合（行がレコードに関連付けられている場合）、**Displayed line number** 関数から返される値は **Selected record number** 関数から返される値と同じです。

Selected record number 関数と同様に、**Displayed line number** 関数から返される値は 1 から始まります。

このコマンドは、空の行も含め、画面上に表示されたリストフォームの各行を処理したい場合に役立ちます。

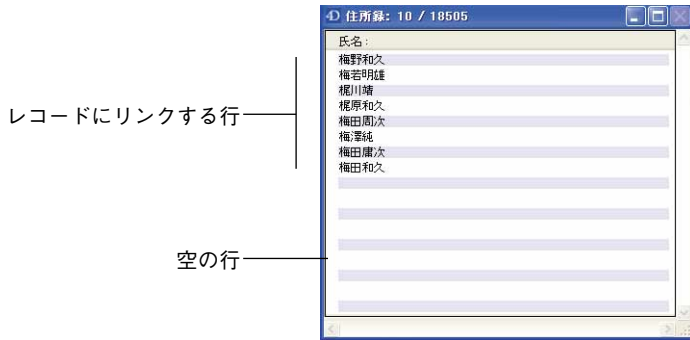
Displayed line number 関数はリスト表示以外の場面で呼び出されると、0 を返します。

▼ 次の例題により、画面上に表示されるリストフォームに対し、レコードが表示されない行に対しても代替色を割り当てることができます。

```

`リストフォームのメソッド
If(Form event=On display detail)
  If(Displayed line number% 2= 0)
    `偶数行のテキストに対しては白地に黒
    SET RGB COLORS([Table 1]Field1; 0; 0x00FFFFFF)
  Else
    `奇数行のテキストに対しては水色地に黒
    SET RGB COLORS([Table 1]Field1; 0; 0x00E0E0FF)
  End if
End if

```



HIGHLIGHT RECORDS

HIGHLIGHT RECORDS({{テーブル;}セット名}{*}))

引数	タイプ	説明
テーブル	テーブル	→ レコードが強調表示されるテーブル 省略した場合、カレントフォームの テーブル
セット名	文字列	→ 強調表示された一連のレコードまたは、 省略した場合は UserSet
*		→ リストの自動スクロールを無効にする

HIGHLIGHT RECORDS コマンドは2つの引数を新たに受け入れます。

引数<テーブル>を使用し、“強調表示”するレコードが属すテーブルを指定します。この新しいパラメータにより、特にカレントテーブルに属さない組み込みサブフォームのレコードを強調表示することができます。

引数<*>を渡すと、強調表示されたレコードが表示されていない場合に、そのリストの自動スクロール機能が無効になります。このメカニズムにより、新しい **SCROLL LINES** コマンドを使用して独自のスクロール管理を行えるようになります。

参照

GET HIGHLIGHTED RECORDS、SCROLL LINES

MODIFY SELECTION、DISPLAY SELECTION

MODIFY SELECTION({テーブル} {; 選択モード} {; リスト更新} {; * {; *}})

DISPLAY SELECTION({テーブル} {; 選択モード} {; リスト更新} {; * {; *}})

引数	タイプ	説明
テーブル	テーブル	→ 表示または変更するテーブル、または省略した場合、デフォルトテーブル
選択モード	倍長整数	→ 選択モード
リスト更新	ブール	→ リスト更新オプションを許可する
*	*	→ 単一レコードの場合に出力フォームを使用し、かつ入力フォームのスクロールバーを隠す
*	*	→ 入力フォームのスクロールバーを表示する (=1番目の<*>引数の後者の機能を無効にする)

これら2つのコマンド、**MODIFY SELECTION**および**DISPLAY SELECTION**は2つの任意の引数を新たに受け入れます。これらの引数を使用し、表示されるフォームに対する選択モードやリストデータの入力オプションを定義することができます。

さらに、これらのコマンドは、「終了」ボタンをデフォルトで表示しなくなりました。

引数<選択モード>を使用すると、レコードリストの選択オプションを設定することができます。この引数には、「Form Options」テーマのいずれかの定数を渡すことができます。

- 「No Selection(0)」(セレクションなし)を渡すと、リストのレコードを選択することはできません。
- 「Single Selection(1)」(1レコード)を渡すと、一度に1レコードだけを選択できます。
- 「Multiple Selection(2)」(複数レコード)を渡すと、Shiftキー(連続したレコードの選択)または、Windows上ではCtrlキー、Mac OS上ではCommandキー(連続しないレコードの選択)を使用して、複数レコードを同時に選択することができます。

引数<選択モード>を渡さない場合は、デフォルトとして「Multiple Selection」オプションが使用されます。

引数<リスト更新>を使用すると、表示されるリストの「リスト更新」オプションを有効にすることができます。このオプションを有効にするには“True”を渡し、無効にするには“False”を渡します。

引数<リスト更新>を渡さない場合は、デフォルトとして「リスト更新」オプションは無効になります。

注：SET ENTERABLE コマンドにより、オンザフライで「リスト更新」オプションを有効、または無効にできるようになりました（前述のSET ENTERABLE コマンドを参照）。

「終了」ボタン

4th Dimension 2004 では、これらのコマンドで使用されるフォームにボタンが存在しない場合でも、リストのフッタ部にデフォルトの「終了」ボタンが追加されません。この際、リストを終了するにはEscape キーだけを使用することができます。

GOTO SELECTED RECORD

GOTO SELECTED RECORD({テーブル;}レコード位置番号)

引数	タイプ	説明
テーブル	テーブル	→ 選択されたレコードに移動するテーブル、または省略された場合はデフォルトテーブル
レコード位置番号	数値	→ セレクション内のレコード位置

GOTO SELECTED RECORD コマンドは、引数<レコード位置番号>として値0を受け入れるようになりました。この値を渡すと、<テーブル>のカレントレコードが存在しなくなります。これにより、リスト上の全レコード、特に組み込みサブフォーム上の全レコードの選択を解除することができます。

Web サーバ

4th Dimension 2004 の「Webサーバ」テーマは、3つの新規コマンド（**PROCESS HTML TAGS**、**GET HTTP BODY**、**SEND HTTP RAW DATA**）によって機能強化されています。また、**GET WEB FORM VARIABLES** コマンドの機能も改良されています。

さらに、CGIとの統合をより向上するために、新規コマンドである **SET CGI EXECUTABLE** が追加されています。

注：新しいSEND HTTP RAW DATA コマンドは、4th Dimension WebサーバによるHTTP/1.1 サポートに関連する一連の新機能の一部です。詳細については、「Webサーバ」の章を参照してください。

PROCESS HTML TAGS

PROCESS HTML TAGS(入力データ; 出力データ)

引数	タイプ	説明
入力データ	テキストまたはBLOB →	処理するHTMLタグを含むデータ
出力データ	テキストまたはBLOB ←	処理されたデータ

PROCESS HTML TAGS コマンドにより、4th Dimension は引数<入力データ> (テキストまたはBLOBタイプのフィールドや変数) に格納された4D HTMLの処理を開始し、結果のデータを<出力データ>に返します。

このコマンドを使用すると、**SEND HTML BLOB** タイプのコマンドを利用したり、またはURLを用いて接尾辞“.shtml”が付加されたページをリクエストすることにより、WebサーバがHTMLページを送信しなくても、タグ付けされたHTMLコードの処理を実行することができます。4th Dimension Webサーバを開始する必要さえありません。

引数<入力データ>には、処理するタグを含むデータを渡します。この引数は、テキストやBLOBタイプのフィールドまたは変数です。BLOBには文字数の制限がないため(テキストタイプの場合は32,000桁まで)、一般的にBLOBタイプを使用する方が望ましいということを覚えておいてください。

Webサーバの処理モードとは関係なく、またWebサーバが開始していなくても、すべての4D HTMLタグが考慮されます(4DVAR、4DSCRIPT、4DLOOPなど)。これらのタグに関する詳細は、4th Dimension『ランゲージリファレンス』マニュアルの「4D HTMLタグ」の節を参照してください。

注：Webサーバ (Webプロセス) の枠外で4DINCLUDEタグを使用する場合、

- ・4th Dimension または4D Serverを使用している場合、デフォルトフォルダはデータベースストラクチャファイルが格納されているフォルダです。
- ・4D Clientを使用している場合、デフォルトフォルダは4D Clientアプリケーションが格納されているフォルダです。

コマンドの実行後、引数<出力データ>には引数<入力データ>のデータが返され、さらに、データに含まれているあらゆる4D HTMLタグの処理結果も一緒に代入されます(該当する場合)。引数<入力データ>に4D HTMLタグがひとつも含まれていない場合、<出力データ>と<入力データ>の内容は同じになります。

引数<出力データ>はフィールドまたは変数のいずれでも構いませんが、引数<入力データ>と同じタイプでなくてはなりません。

このコマンドにより、データベースにあるHTMLタグの処理結果である値が送信される前に、それを保存しておくことができます。

また、このコマンドはWebサーバの使用だけではなく、4D HTMLタグの解析も行えます。特に、コマンドを用いて、データベース内のデータに関する処理手続きや、その参照を含む電子メールメッセージを4D Internet Commandsを介してHTMLフォーマットで送信することができます。

▼ 次の例題は、このコマンドの機能を示しています。

```
C_BLOB($in)
C_BLOB($out)
C_TEXT($in_text)
C_TEXT(Var)
C_TEXT(HTMLvar)
Var:="<B>"
$in_text:="<p><!--#4DVAR Var--></p>"
TEXT TO BLOB($in_text;$in;Text without length)
PROCESS HTML TAGS($in;$out)
HTMLvar:=BLOB to text($out;Text without length)
`HTMLvarには <p>&#60;B&#62;</p> が格納される
```

GET HTTP BODY

GET HTTP BODY(ボディ)

引数	タイプ	説明
ボディ	BLOBまたはText	← HTTPリクエストのボディ部

GET HTTP BODY コマンドは、処理されるHTTPリクエストのボディ部 (body) を返します。HTTPリクエストのボディは、処理や解析が行われずにそのままの状態で見返されます。

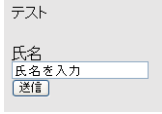
このコマンドは非コンテキストモードでのみ動作し、Webデータベースメソッド (**On Web Authentication**、**On Web Connection**) または非コンテキストモードで実行される任意のWebメソッドから呼び出すことができます。

<ボディ>には、BLOBまたはテキストタイプのフィールドや変数を渡すことができます。BLOBには文字数の制限がないため、一般的にBLOBタイプを使用の方が望ましいということ覚えておいてください。これに反してテキストタイプの場合は、32,000桁までに制限され、この桁数を超えると、受信されたデータのうち超過した部分が切り捨てられます。

例えば、このコマンドを使用して、リクエストのボディ部でクエリを実行することができます。また、上級ユーザの場合は、4th Dimensionデータベース内でWebdevサーバを設置することも可能です。

▼この例題では、4th Dimension Webサーバへシンプルなリクエストを送信し、デバッグ上にHTTPリクエストのボディの内容を表示します。

4th Dimension Webサーバへ送信されたフォームならびに対応するHTMLコードを次に示します。

フォーム	ボディ
	<pre> <HTML> <HEAD> <TITLE>テストページ</TITLE> <META http-equiv="content-type" content="text/html;charset=shift_JIS"> </HEAD> <BODY> テスト <FORM ACTION="/4DACTION/Web_test" METHOD="post"> <input type="text" value="氏名を入力" name="vName"> <input type="submit"> </FORM> </BODY> </HTML> </pre>

次に示すのはTest4D2004 メソッドです。

```

C_BLOB($request)
C_TEXT($requestText)
GET HTTP BODY($request)
$requestText:=BLOB to text($request;Text without length)
SEND HTML FILE("page.html")

```

注：このメソッドに対して「4DACTIONで使用可能」プロパティが定義されています。

このフォームをWebサーバへサブミットすると、変数\$requestTextにはHTTPリクエストのボディのテキストが返されます。

SEND HTTP RAW DATA

SEND HTTP RAW DATA(データ{;*})

引数	タイプ	説明
データ	BLOB →	送信する HTTP データ
*	*	→ “chunked (塊)” 送信

このコマンドを使用して、4D Webサーバは“raw (ロウ)” HTTPデータを送信することができます。また、このデータは“chunked (塊)”形式にすることができます。このコマンドは非コンテキストモードでのみ動作します。

引数<データ>には、HTTPレスポンスの2つの標準部分、つまりヘッダ部とボディ部を格納します。データは、サーバによる事前のフォーマット処理が行われなまま送信されます。しかし、4th Dimensionはレスポンスのヘッダとボディに関する簡単なチェックを実行して、これらのデータが有効であることを確認します。

- ヘッダが不完全であったり、HTTP プロトコル仕様に準拠していない場合、4th Dimension は適宜に変更を行います。
- HTTP リクエストが不完全である場合、4th Dimension は不足している情報を追加します。例えば、リクエストのリダイレクトを行いたい場合は、次のように記述しなくてはなりません。

```
HTTP/1.1 302
Location:http://...
```

しかし、次の文だけを渡した場合、

```
Location:http://...
```

4th Dimension は “HTTP/1.1 302” を付加してリクエストを完成させます。

任意の引数 < * > を使用して、レスポンスを “chunked (塊) 形式” で送信するように指定することができます。サーバがその合計長を認識せずにレスポンスを送信する場合は、レスポンスをチャンクに分けると便利です (例えば、レスポンスがまだ生成されていない場合)。HTTP/1.1 に対応するブラウザはすべてチャンク形式のレスポンスを受け入れません。

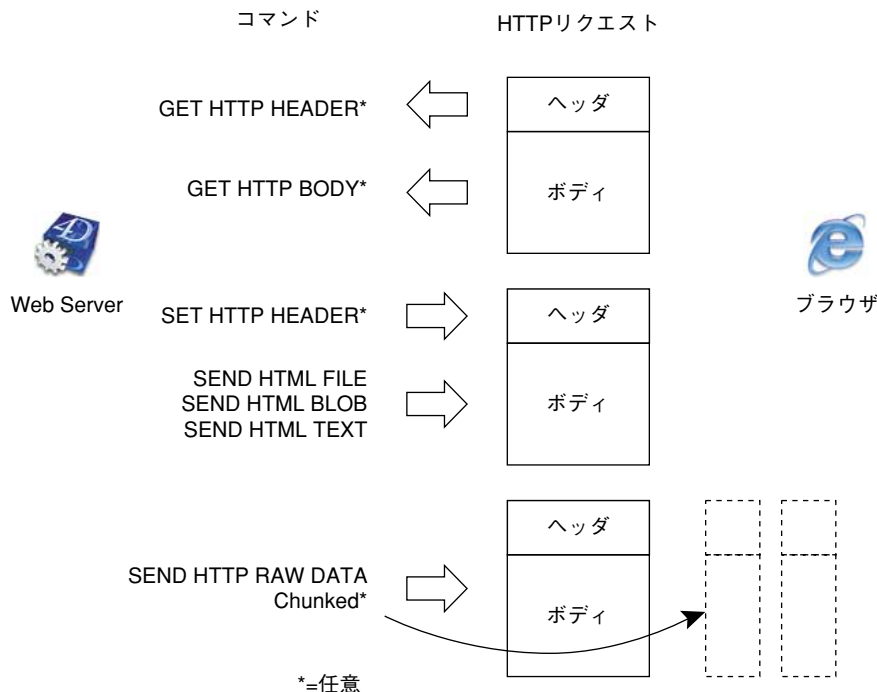
引数 < * > を渡すと、必要に応じて Web サーバは自動的に 「transfer-encoding: chunked (チャンク形式エンコーディング)」 フィールドをレスポンスのヘッダに組み込みます (希望する場合は、レスポンスヘッダを手動で処理することが可能)。レスポンスの残りの部分もまた、“chunked” オプションのシンタックスに合うようにフォーマットされます。“chunked” 形式のレスポンスには、1つのヘッダと不定数のボディのチャンクが納められます。同一メソッド内において **SEND HTTP RAW DATA(data;*)** の実行に続けて行われるすべての **SEND HTTP RAW DATA** 命令は、レスポンスの一部とみなされます (引数 < * > が指定されているかどうかに関わらず)。このメソッドの実行が完了すると、サーバは “chunked” 形式の送信を終了します。

注： Web クライアントが HTTP/1.1 をサポートしていない場合、4th Dimension はそのレスポンスを自動的に HTTP/1.0 互換フォーマットへ変換します (送信されるデータは “chunked” 形式にはならない)。ただし、この場合は望ましい結果にならない可能性があります。したがって、Web ブラウザが HTTP/1.1 をサポートするかどうかを調べて、適切なレスポンスを送信することをお勧めします。これを実現するため、次のようなコードを使用することができます。

```
C_BOOLEAN($0)
ARRAY TEXT(arFields;0)
ARRAY TEXT(arValues;0)
GET HTTP HEADER(arFields;arValues)
$0:=False
If(Size of array(arValues)>=3)
```

```
If(Position("HTTP/1.1";arValues{3})>0)
  $0:=True `ブラウザは HTTP/1.1 をサポートする
End if
End if
```

このコマンドを **GET HTTP BODY** コマンドや「Webサーバ」テーマの他のコマンドと組み合わせて使用すると、4D 開発者が利用できるツールの種類がすべて揃い、受信側と送信側の HTTP 接続の処理を完全にカスタマイズすることができます。これらの各種ツールを次の図に示します。



▼この例題は、**SEND HTTP RAW DATA** コマンドとともに“chunked”オプションを使用する例を示しています。ループにおいて処理中に生成された100個のチャンクとしてデータが送信されます。レスポンスのヘッダは明示的に設定されないということを覚えておいてください。つまり、引数<*>が使用されているため、**SEND HTTP RAW DATA** コマンドはヘッダを自動的に送信し、「transfer-encoding:chunked」フィールドをその中に挿入します。

```
C_LONGINT($cpt)
C_BLOB($my_blob)
C_TEXT($output)
For($cpt;1;100)
```

```
$output:="[ "+String($cpt)+"]"  
TEXT TO BLOB($output;$my_blob;Text without length)  
SEND HTTP RAW DATA($my_blob;*)  
End for
```

GET WEB FORM VARIABLES

4th Dimension 2004では、このコマンドの処理機能が拡張されています。Webサーバへ送信されるURLのタイプとは関係なく、このコマンドを利用できるようになりました。具体的に言うと、すべてのURLへPOSTデータを送信するHTMLフォームで動作できるようになります。

以前のバージョンの4th Dimensionでは、“/4DACTION”、“/4DMETHOD”、“/4DCGI”で開始するリクエスト、またはリクエスト文字列を含むリクエストだけがこのコマンドで解析されていました。

CGIのサポート

4th Dimension 2004では、CGIのサポートがシンプルになり、拡張されています。4th Dimension Webサーバは、Mac OS XおよびWindowsの双方において、あらゆるタイプのCGIを呼び出せるようになりました。

注：今までの4th Dimensionと4D WebSTAR間のCGI通信メカニズム（Apple Eventsの利用に基づく）は、4th Dimension 2004ではサポートされなくなります。

4th Dimension 2004におけるCGIサポートの変更により、2つの新規コマンドが追加されました。

■ SET CGI EXECUTABLE

■ SET ENVIRONMENT VARIABLE（「システム環境」テーマ）

このコマンドの有効範囲は4th Dimension Webサーバに限定されていないため、コマンドは「システム環境」テーマに置かれています。

SET CGI EXECUTABLE

SET CGI EXECUTABLE(url1{; url2})

引数	タイプ		説明
url1	文字列	→	アクセスURL
url2	文字列	→	アクセスURL

SET CGI EXECUTABLE コマンドは、Web ユーザに対して URL 上に CGI を表示しないで、それを実行するために使用します。

特に、このコマンドは「On Web Authentication」データベースメソッドにおいて、例えば、実行する CGI を決定するために使用されます。また、Mac OS X および Windows 双方で動作します。

< url1 > には、実行する CGI のアクセス URL を渡します。例えば、**SET CGI EXECUTABLE("/myfile.pl")** という命令を記述した場合、4D Web サーバは CGI 「myfile.pl」を実行します。その際、このアプリケーションを Web サーバのデフォルトフォルダ内に配置しなくてはなりません。

< url1 > に空の文字列 ("") を渡すと、4th Dimension はブラウザから直接送信された URL 内に指定された CGI を実行します (適用可能な場合)。

任意の引数 < url2 > には、その CGI で処理したいファイルのアクセス URL を渡します。例えば、**SET CGI EXECUTABLE("cgi-bin/Perl2.cgi";"Perl2.pl")** という命令を記述した場合、Web サーバは「Perl2.pl」ファイルを渡した上で、CGI である「Perl2.cgi」を実行します。

< url2 > に空の文字列 ("") を渡すと、4th Dimension はブラウザから送信された URL に指定されたファイルを CGI へ渡して処理します。特に PHP でこのメカニズムが頻繁に用いられます (例: **SET CGI EXECUTABLE("/cgi-bin/php";"")**)。

コマンドにより指定されたアクセス URL が正しくない場合、ブラウザは “File not found” エラーページを表示します。

ただし、**SET CGI EXECUTABLE** コマンドはエラーを直接返さないということを覚えておいてください。このコマンドは CGI がコールされた時に次に使用される “現在値” を設定するだけです。このコマンドを何度も呼び出した場合は、一番最後の呼び出しで指定された値が使用されます。

▼ この例題では、「cgi-bin」フォルダに置かれていない「example.php」ファイルが、「cgi-bin」フォルダ内の CGI 「Perl2.cgi」によって処理されます。

```
SET CGI EXECUTABLE("/cgi-bin/Perl2.cgi";"example.php")
```

配列

「配列」テーマには2つの新規コマンドが追加され、配列内において任意の値が出現した回数を集計したり、配列の最後に項目を追加できるようになりました。

Count in array

Count in array(配列; 値) → 倍長整数

引数	タイプ	説明
配列	配列	→ 集計を行う配列
値	4D表記	→ 集計する値
戻り値	倍長整数	← 出現回数

Count in array関数は、<配列>内における<値>の出現回数を返します。

このコマンドは、テキスト、文字、数値、日付、ポインタ、ブールタイプの配列に対して使用できます。引数<配列>と<値>は同じタイプか、または互換性があるタイプでなくてはなりません。

<値>と一致する項目が<配列>内に存在しない場合、コマンドは0を返します。

▼ 次の例題を使用し、リストボックス内で選択された行の数を表示することができます。

```
`tBList はリストボックス列配列の名前です
ALERT(String(Count in array(tBList,True))+
        "行がリストボックス内で選択されました。")
```

APPEND TO ARRAY

APPEND TO ARRAY(配列; 値)

引数	タイプ	説明
配列	配列	→ 項目を追加する配列
値	4D表記	→ 追加する値

APPEND TO ARRAY コマンドは、<配列>の最後に新規項目を追加し、その項目に<値>を割り当てます。インタプリタモードにおいて、<配列>が存在しない場合、コマンドは<値>のタイプに従って配列を作成します。

このコマンドはあらゆるタイプの配列（文字列、数値、ブール、日付、ポインタ、ピクチャ）に対応します。

<値>のタイプは配列のタイプと一致しなくてはなりません。一致しない場合は、シンタックスエラー 54 “引数のタイプが違います” が生成されます。ただし、次の値は受け入れられます。

■ 文字列タイプの<配列>（テキストまたは文字列）はテキストや文字列タイプの<値>を受け入れます。

■ 数値タイプの<配列>（整数、倍長整数、実数）は、整数、倍長整数、実数、または時間タイプの<値>を受け入れます。

▼ 次のコードの場合

```
INSERT ELEMENT($myarray;Size of array($myarray)+1)
$myarray{Size of array($myarray)}:=$myvalue
```

次の命令に書き換えることができます。

```
APPEND TO ARRAY($myarray;$myvalue)
```

通信

「通信」テーマには、シリアルポートマッピングを処理するための新規コマンドが追加されています。

GET SERIAL PORT MAPPING

GET SERIAL PORT MAPPING(番号配列;名前配列)

引数	タイプ	説明
番号配列	数値配列	← ポート番号の配列
名前配列	文字列配列	← ポート名の配列

GET SERIAL PORT MAPPING コマンドは2つの配列を返し、現行のマシンのシリアルポート番号とシリアルポート名をそれぞれに代入します。

Mac OS Xでは、USBシリアルアダプターの使用時に、OSがポート番号を動的に割り当てるため、このコマンドが役立ちます。シリアルポートの実際のIDとは関係なく、その名前（固定）を用いてシリアルポートを取り扱うことができます。

バックアップ

4th Dimension 2004では、新しいバックアップモジュールに対して使用するランゲージコマンドが「バックアップ」テーマに追加されています（**BACKUP**、**RESTORE**、**GET BACKUP INFORMATION**、**GET RESTORE INFORMATION**）。

また、定期的バックアップの管理のために、「On backup startup」と「On backup shutdown」という2つのデータベースメソッドが新たに追加されています。

このモジュールの機能に関する詳細は、前述の「統合されたバックアップモジュール」の章を参照してください。

BACKUP

BACKUP

引数	タイプ	説明
----	-----	----

このコマンドに引数は必要ありません。

BACKUP コマンドは、現在のバックアップ設定を用いて、データベースのバックアップを開始します。確認ダイアログボックスは表示されませんが、進捗バーが画面上に現れます。

バックアップ設定は、アプリケーションの「環境設定」で設定します。これらの設定は、データベースの「Preferences/backup」サブフォルダ内にある「Backup.XML」ファイルにも保存されます。

BACKUP コマンドは、実行前に「**On backup startup**」データベースメソッドを呼び出し、実行終了時に「**On backup shutdown**」データベースメソッドを呼び出します。

このようなメカニズムなので、このコマンドを「**On backup startup**」と「**On backup shutdown**」データベースメソッドからコールすべきではありません。

4D Server：クライアントマシンから呼び出された場合、**BACKUP** コマンドはストアドプロシージャとみなされ、サーバー上で実行されます。

バックアップが正常に終了すると、システム変数OKには値1が代入され、それ以外の場合は値0が代入されます。障害が発生した場合はエラーが生成されます。このエラーは**ON ERR CALL** コマンドにより設定されたエラー処理用メソッドを用いて遮ることができます。

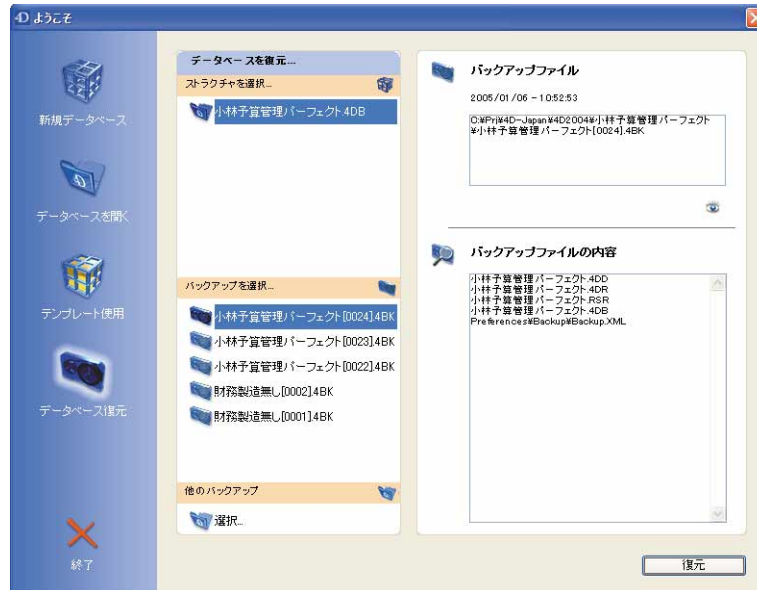
RESTORE

RESTORE

引数	タイプ	説明
----	-----	----

このコマンドに引数は必要ありません。

RESTORE コマンドは4th Dimensionの「ようこそ」ダイアログボックスの「復元」ページを表示します。



ユーザはここで復元するアーカイブを選択することができます。

このコマンドは、カスタマイズされたインターフェースを使用してバックアップを管理する際に役立ちます。

GET BACKUP INFORMATION

GET BACKUP INFORMATION(セレクトタ; 情報1; 情報2)

引数	タイプ	説明
セレクトタ	倍長整数	→ 取得する情報のタイプ
情報1日付: 整数	←	セレクトタの値1
情報2日付: 整数	←	セレクトタの値2

GET BACKUP INFORMATION コマンドを使用し、データベースのデータに関して行われた前回のバックアップに関連する情報を取得することができます。

<セレクトタ>には取得する情報タイプを渡します。この引数の値として、「Backup」テーマ内にある次の定数のいずれかを使用することができます。

定数	タイプ	値
Last Backup Date	倍長整数	0
Last Backup Status	倍長整数	2
Next Backup Date	倍長整数	4

引数<情報1>と<情報2>のタイプおよび内容は、<セレクトタ>の値によって決まります。

- <セレクトタ>=0：<情報1>には前回のバックアップ日付、<情報2>には前回のバックアップ時間が返されます。
- <セレクトタ>=2：<情報1>には前回のバックアップステータスの数値、<情報2>にはそのテキストが返されます。
- <セレクトタ>=4：<情報1>には次回の定期バックアップ日付、<情報2>には次回の定期バックアップ時間が返されます。

GET RESTORE INFORMATION

GET RESTORE INFORMATION(セレクトタ; 情報1; 情報2)

引数	タイプ	説明
セレクトタ	倍長整数	→ 取得する情報のタイプ
情報1日付: 整数	←	セレクトタの値1
情報2日付: 整数	←	セレクトタの値2

GET RESTORE INFORMATION コマンドを使用し、前回の復元に関連する情報を取得することができます。

<セレクトタ>には取得する情報タイプを渡します。この引数の値として、「Backup」テーマ内にある次の定数のいずれかを使用することができます。

定数	タイプ	値
Last Restore Date	倍長整数	0
Last Restore Status	倍長整数	2

引数<情報1>と<情報2>のタイプおよび内容は、<セレクトタ>の値によって決まります。

- <セレクトタ>=0：<情報1>には前回のデータベース復元日付、<情報2>には前回のデータベース復元時間が返されます。
- <セレクトタ>=2：<情報1>には前回のデータベース復元のステータスを表わす数値が、<情報2>にはそのテキストが返されます。

「On Backup Startup」データベースメソッド

この新しいデータベースメソッドは、データベースのバックアップを開始しようとするたびに呼び出されます（手動でのバックアップ、定期的自動バックアップ、または **BACKUP** コマンドによるバックアップ）。

これは、すべての4D環境、つまり4th Dimension、4D Server、4D Client、4D Runtime、ならびに4D Engineが組み込まれたデータベースに関係します。

「**On Backup Startup**」データベースメソッドを使用すると、開始されたバックアップを検証することができます。このメソッドにおいて、バックアップの許可または拒否を示す値を引数\$0に返してください。

■ \$0=0：バックアップを開始することができます。

■ \$0#0：バックアップは許可されません。処理はキャンセルされ、エラーが返されます。このエラーは、**GET BACKUP INFORMATION** コマンドを使用して取得することができます。

このデータベースメソッドを使用すると、バックアップの実行状況を確認することができます（ユーザ、前回のバックアップ日付など）。

引数\$0（倍長整数タイプ）は、データベースメソッド内で宣言しなくてはなりません。

C_LONGINT(\$0)

「On backup shutdown」データベースメソッド

この新しいデータベースメソッドは、データベースのバックアップが終了するたびに呼び出されます。バックアップが中止される原因として、コピーの終了やユーザによる中断、エラーがあります。

これは、すべての4D環境、つまり4th Dimension、4D Server、4D Client、4D Runtime、ならびに4D Engineが組み込まれたデータベースに関係します。

「**On backup shutdown**」データベースメソッドを使用すると、バックアップが正常に実行されたかどうかを確認できます。バックアップが完了すると、このメソッド内の引数\$1にはバックアップのステータスを示す値が返されます。また、バックアップが正常に終了すると、引数\$1には0が代入されます。

他にも、「**On Backup Startup**」データベースメソッド（\$0#0）によりバックアップが拒否された場合は、引数\$0へ実際に返された値が引数\$1に代入されます。これにより、独自のエラー管理システムを実装することができます。

引数\$1 (倍長整数タイプ) は、データベースメソッド内で宣言しなくてはなりません。

C_LONGINT(\$1)

XML

4th Dimension 2004 では、XML (eXtensible Markup Language) 構造をサポートするためにさらに多くのコマンドやツールが提供されています。主に3つの開発分野が支援されています。

■ SAX 標準のサポート

■ XML 構造の記述や Xpath 表記を取り扱うために使用されるコマンドの追加。

■ XSL および Xpath 言語のサポート

4D 2003 におけるコマンド名の変更

SAX 標準の統合や、各コマンドの用法を明確にする目的のために、4th Dimension 2003 の XML コマンド名の大半が変更されています。これらコマンド名の先頭には「DOM」が付けられていますが、その働きは変わりません。

以前のコマンド名 (4th Dimension 2003.x) 新しいコマンド名 (4th Dimension 2004.x)

Parse XML source	DOM Parse XML source
Parse XML variable	DOM Parse XML variable
Count XML elements	DOM Count XML elements
Get First XML element	DOM Get First XML element
Get Next XML element	DOM Get Next XML element
Get XML element	DOM Get XML element
GET XML ELEMENT NAME	DOM GET XML ELEMENT NAME
GET XML ELEMENT VALUE	DOM GET XML ELEMENT VALUE
Count XML attributes	DOM Count XML attributes
GET XML ATTRIBUTE BY INDEX	DOM GET XML ATTRIBUTE BY INDEX
GET XML ATTRIBUTE BY NAME	DOM GET XML ATTRIBUTE BY NAME
GET XML ERROR	GET XML ERROR(unchanged)
Parse XML information	DOM Parse XML information
CLOSE XML	DOM CLOSE XML

SAX 標準のサポート

4th Dimension 2004 ではXML SAX標準がサポートされるため、サイズ制限なしでXML文書処理できるようになりました。

SAX とは？

DOM (Document Object Model) およびSAX (Simple API XML) は、XML文書用の2つの解析モードです。

- DOMモードは、XMLソースを解析し、その構造(“ツリー(木)”)をメモリ上に作成します。このため、ソースの各要素へのアクセスが非常に高速に行われます。しかし、ツリー構造全体がメモリ上に保持されているため、大規模なXML文書の処理によりメモリ容量を超えてしまい、エラーが発生する可能性があります。
- SAXモードは、ツリー構造をメモリ上に作成しません。このモードでは、ソースの解析時に“イベント”(要素の初めと終わりなど)が生成されます。このモードにより、利用可能なメモリ量とは関係なく、あらゆるサイズのXML文書を解析することができます。

XML規格に関する詳細は、次のサイトを参考してください。

<http://www.saxproject.org/?selected=event>

<http://www.w3schools.com/xml/>

SAX を介したXML文書の作成、オープン、クローズ

SAXコマンドは、4th Dimension標準のドキュメント参照(DocRef、時間タイプの参照)を用いて動作します。したがって、**SEND PACKET**や**Append document**などの文書管理用4Dコマンドと一緒に、これらのコマンドを使用することができます。

プログラムからのXML文書の作成やオープンには、**Create document**関数および**Open document**関数を使用して行います。その後、これらのコマンドと一緒にXMLコマンドを使用すると、符号化などのXMLメカニズムが自動的に実装されます。例えば、`<?xml version="1.0" encoding="...encoding..." standalone="no"?>`というヘッダが自動的に文書へ記述されます。

XML文書のクローズは、**CLOSE DOCUMENT**コマンドを使用して行わなければなりません。オープンされているXML要素が存在すれば、それも自動的にクローズされます。

SAX SET XML OPTIONS

SAX SET XML OPTIONS(文書; エンコード{; スタンドアロン{; インデント}})

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
エンコード	文字列	→ XML文書の文字セット
スタンドアロン	ブール	→ True=文書はスタンドアロン False (デフォルト) =文書はスタンドアロンではない
インデント	ブール	→ True (デフォルト) =文書はインデントされる False=文書はインデントされない

SAX SET XML OPTIONS コマンドは、引数に渡された値を用いて、<文書>で参照されるXML文書を初期化します。これらの引数を使用し、エンコード、スタンドアロン属性、ならびに文書のインデントを決定することができます。

■ **エンコード (符号化方式)** : 文書で使用する文字セットを示します。デフォルトとして (このコマンドが呼び出されない場合)、UTF-8文字セット (圧縮Unicode) が使用されます。

■ **スタンドアロン** : 文書が独立しているか (True)、または動作するために他のファイルや外部リソースを必要とするか (False) を示します。デフォルトでは (このコマンドが呼び出されない場合や引数が省略された場合)、ドキュメントは独立していません。

■ **インデント** : XMLキーの階層構造に従って文書のインデントを表示するか (True)、表示しないか (False) を示します。デフォルトでは (このコマンドが呼び出されない場合や引数が省略された場合)、文書はインデントされます。

このコマンドは、文書内の最初のセットコマンドの前、XML文書毎に一度呼び出さなくてはなりません。これを行わないと、エラーメッセージが生成されます。

▼ 次の命令の場合

```
SAX SET XML OPTIONS($RefDoc;"UTF-16";True)
```

文書内には、次の行が作成されます。

```
<?xml version="1.0"encoding="UTF-16"standalone="yes"?>
```

SAX ADD XML DOCTYPE

SAX ADD XML DOCTYPE(文書; 文書タイプ)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
文書タイプ	文字列	→ 追加される DocType

SAX ADD XML DOCTYPE コマンドは、<文書>で参照される XML 文書に DocType 命令を追加します。

DocType 命令により、文書が書き込まれた XML のタイプを示し、使用される Document Type Declaration (DTD) を指定することができます。一般的に、DocType 命令は次の形式になります。

```
<!DOCTYPE XML_type"DTD_address>
```

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、それ以外の場合は 0 が代入されます。エラーの場合、コマンドはエラーを返し、このエラーはエラー処理メソッドを用いて遮ることができます。

▼ 次の命令の場合

```
vDocType:= "SYSTEM Books¥"Book.DTD¥"
SAX ADD XML DOCTYPE($RefDoc;vDocType)
```

文書内には、次の行が作成されます。

```
<!DOCTYPE SYSTEM Books"Book.DTD>
```

SAX ADD XML COMMENT

SAX ADD XML COMMENT(文書; コメント)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
コメント	文字列	→ 追加されるコメント

SAX ADD XML COMMENT コマンドは、<文書>で参照される XML 文書にコメントを追加します。

XML コメントはテキストであり、その内容は XML インタプリタによって解析されません。XML コメントは “<!--” と “-->” 文字で囲まなくてはなりません。

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、それ以外の場合は 0 が代入されます。エラーの場合、コマンドはエラーを返し、このエラーはエラー処理メソッドを用いて遮ることができます。

▼ 次の命令の場合

```
vComment:= "Created by 4th Dimension"  
SAX ADD XML COMMENT($RefDoc;vComment)
```

文書内には、次の行が作成されます。

```
<!--Created by 4th Dimension-->
```

SAX OPEN XML ELEMENT

SAX OPEN XML ELEMENT(文書; タグ{; 属性名; 属性値} ...{;属性名 X;属性値 X})

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
タグ	文字列	→ オープンする要素の名前
属性名	文字列	→ 属性の名前
属性値	文字列	→ 属性の値

SAX OPEN XML ELEMENT コマンドを使用し、<文書>により参照されるXML文書に新しい要素を追加し、さらにオプションとして属性とその値を追加することができます。

追加された要素は文書内において“オープン”な状態です（終了タグが追加されていない）。このコマンドを使用して作成された要素をクローズするには、次のいずれかの処理を行います。

■ **SAX CLOSE XML ELEMENT** コマンドを使用します。または、

■ XML文書をクローズします。この場合、必要に応じて4th Dimensionは自動的にクローズXMLタグを付加します。

<タグ>には、作成される要素の名前を渡します。この名前には、文字、数値、“.”、“-”、“_”、“:”などの記号だけを含めることができます。無効な文字を<タグ>に渡すと、エラーが生成されます。

オプションとして、コマンドは引数<属性名>と<属性値>を用いて1つ以上の属性と値のペアを渡すことができます（変数やフィールド、リテラル値の形で）。必要な数の属性と値のペアを渡すことができます。

▼ 次の命令の場合、

```
vElement:= "Book"  
SAX OPEN XML ELEMENT($RefDoc;vElement)
```

文書内には、次の行が作成されます。

```
<Book>
```

SAX OPEN XML ELEMENT ARRAYS

SAX OPEN XML ELEMENT ARRAYS(文書; タグ{; 属性名配列;属性値配列}...{; 属性名配列 X;属性値配列 X})

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
タグ	文字列	→ オープンする要素の名前
属性名配列	文字列配列	→ 属性名の配列
属性値配列	文字列配列	→ 属性値の配列

SAX OPEN XML ELEMENT ARRAYS コマンドを使用すると、<文書>に渡された参照番号を持つXML文書に新しい要素を追加し、さらにオプションとして配列形式で提供される属性とその値を追加することができます。

配列がサポートされる点を除けば、このコマンドは**SAX OPEN XML ELEMENT** コマンドと同じです。このコマンドの動作に関する詳細については、**SAX OPEN XML ELEMENT** コマンドの説明を参照してください。

オプションとして、**SAX OPEN XML ELEMENT ARRAYS** コマンドは、引数<属性名配列>と<属性値配列>の配列を用いて属性と値のペアを渡すことができます。この配列は事前に作成し、属性と属性値を対で使用しなくてはなりません。配列のペアおよび各ペア内の項目は、必要な数だけ渡すことができます。

▼ 次の4Dコードの場合

```

ARRAY STRING(80;tAttrNames;2)
ARRAY STRING(80;tAttrValues;2)
vElement:="Book"
tAttrNames{1}:="Font"
tAttrValues{1}:="Arial"
tAttrNames{2}:="Style"
tAttrValues{2}:="Bold"
SAX OPEN XML ELEMENT ARRAYS($RefDoc;vElement;tAttrNames;tAttrValues)

```

文書内には、次の行が作成されます。

```
<Book Font="Arial"Style="Bold">
```

SAX CLOSE XML ELEMENT

SAX CLOSE XML ELEMENT(文書)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号

<文書>で参照されるXML文書において、**SAX CLOSE XML ELEMENT** コマンドは、**SAX OPEN XML ELEMENT** コマンドで前回オープンされた要素のクローズに必要な命令を記述します。

このコマンドの使用は任意です。実際、4th DimensionはXML文書のクローズ時に、閉じられていない要素に対して必要な終了タグを自動的に追加します。

▼ 前回オープンされた要素が<Book>である場合、次の命令を実行します。

SAX CLOSE XML ELEMENT(\$RefDoc)

文書内には、次の行が作成されます。

</Book>

SAX ADD XML ELEMENT VALUE

SAX ADD XML ELEMENT VALUE(文書; データ)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
データ	テキスト!BLOB	→ 文書に挿入するテキストまたはBLOB

SAX ADD XML ELEMENT VALUE コマンドは、<文書>で参照されるXML文書内に、テキストやBLOBタイプの<データ>を変換せずに直接追加します。このコマンドは、例えば、電子メールの本文に添付書類を挿入する動作と同じです。

<データ>の内容を符号化したい場合は、**ENCODE** コマンドを使用しなくてはなりません。その際、もちろん<データ>にはBLOBを渡さなくてはなりません。

このコマンドが正しく動作するために、要素は必ずオープンな状態にしておいてください。オープンな状態でなければ、エラーが生成されます。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、それ以外の場合は0が代入されます。

▼ この例題は、開かれているXML要素内にwhitepaper.pdfファイルを挿入します。

C_BLOB(vBMyBLOB)

DOCUMENT TO BLOB("c:II whitepaper.pdf";vBMyBLOB)

SAX ADD XML ELEMENT VALUE(\$RefDoc;vBMyBLOB)

SAX ADD XML CDATA

SAX ADD XML CDATA(文書; データ)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
データ	テキスト; BLOB	→ 文書内のCDATAタグの間に挿入するテキストまたはBLOB

SAX ADD XML CDATA コマンドは、<文書>で参照されるXML文書内へテキストやBLOBタイプの<データ>を追加します。この<データ>は、自動的に<CDATA>と</CDATA>タグの間に入れます。CDATAセクションに組み込まれたテキストは、XMLインタプリタより無視されます。

<データ>の内容を暗号化したい場合は、**ENCODE** コマンドを使用しなくてはなりません。その際はもちろん、<データ>に**BLOB**を渡さなくてはなりません。

このコマンドが正しく動作するために、要素は必ずオープンな状態にしておいてください。オープンな状態でなければ、エラーが生成されます。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、それ以外の場合は0が代入されます。

▼ XML文書に次の行を挿入したいものとします。

```
function matchwo(a,b)
{
    if(a< b&& a< 0) then
    {
        return 1
    }
else
    {
        return 0
    }
}
```

これを行うには、次のコードを実行するだけです。

```
C_TEXT(vtMytext)
`変数 vtMytext のテキストをここに配置する
SAX ADD XML CDATA($RefDoc;vtMytext)
```

結果は次のようになります。

```
<![CDATA[
function matchwo(a,b)
```

```

{
if(a<b&& a<0)then
    {
    return 1
    }
else
    {
    return 0
    }
}
]]>

```

SAX ADD PROCESSING INSTRUCTION

SAX ADD PROCESSING INSTRUCTION(文書; 命令)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
命令	テキスト	→ 文書に挿入するテキストまたはBLOB

SAX ADD PROCESSING INSTRUCTION コマンドは、<文書>で参照されるXML文書内へXML処理手続きである<命令>を追加します。

処理命令により、アプリケーションタイプを指定したり、必要な場合は未処理の外部エンティティを処理する追加パラメータを指定することができます。

このコマンドは、XMLに準拠して命令文のデータの書式を整えます。しかし、命令文そのものは解析されないため、開発者の責任において構文が有効であることを確認してください。

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 次のコードの場合

```

vtStatement:="xml-stylesheet type="+Char(Quote)+"text/xsl"+
Char(Quote)+ "href="+Char(Quote)+
"style.xml"+Char(Quote)
SAX ADD PROCESSING INSTRUCTION($RefDoc;vtStatement)

```

文書内には、次の行が作成されます。

```
<?xml-stylesheet type="text/xsl"href="style.xml"?>
```


SAX Get XML node

SAX Get XML node(文書) → 倍長整数

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
戻り値	倍長整数	← 関数から返されるイベント

このコマンドは倍長整数を返し、〈文書〉で参照されるXML文書の解析時に返されたSAXイベントのタイプを示します。

返されるイベントの値は、「XML」テーマの定数として入手することができます。

定数	タイプ	値
XML Start Document	倍長整数	1
XML Comment	倍長整数	2
XML Processing Instruction	倍長整数	3
XML Start Element	倍長整数	4
XML End Element	倍長整数	5
XML DATA	倍長整数	6
XML CDATA	倍長整数	7
XML Entity	倍長整数	8
XML End Document	倍長整数	9

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 次の例題は任意のイベントを処理します。

```

$RefDoc:=Open document("";"";Read Mode)
If (OK=1)
  Repeat
    $MyEvent:=SAX Get XML node($RefDoc)
  Case of
    ¥ ($MyEvent=XML Start Document)
      `処理1を実行
    ¥ ($MyEvent=XML Comment)
      `処理2を実行
  End case
Until($MyEvent=XML End Document)
End if
CLOSE DOCUMENT($RefDoc)

```

SAX GET XML DOCUMENT VALUES

SAX GET XML DOCUMENT VALUES(文書; エンコード; バージョン, スタンドアロン)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
エンコード	文字列	← XML文書の文字セット
バージョン	文字列	← XMLバージョン
スタンドアロン	ブール	← True=文書は独立している、それ以外の場合はFalse

SAX GET XML DOCUMENT VALUES コマンドは、<文書>で参照されるXML文書のXMLヘッダから基本情報を取得します。

コマンドは、符号化方式のタイプ、バージョン、文書の“independent”属性を引数<エンコード>、<バージョン>、<スタンドアロン>にそれぞれ返します。このコマンドはSAXイベントである「XML Start Document」と一緒に使用しなくてはなりません。

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、それ以外の場合は0が代入されてエラーが返されます。

SAX GET XML COMMENT

SAX GET XML COMMENT(文書; コメント)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
コメント	文字列	← XMLコメント

このコマンドは、<文書>で参照されるXML文書においてXML CommentタイプのSAXイベントが生成されると、コメントを返します。

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、それ以外の場合は0が代入されてエラーが返されます。

SAX GET XML PROCESSING INSTRUCTION

SAX GET XML PROCESSING INSTRUCTION(文書; 名前; 値)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
名前	文字列	← 命令の名前
値	文字列	← 命令の値

このコマンドは、<文書>で参照される XML 文書において処理された XML 命令の<名前>と<値>を返します。このコマンドは、「XML Processing Instruction」イベントと一緒に呼び出さなくてはなりません。

▼ 次の XML コードについてみてみましょう。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Edited with XML Spy v3.0.7 NT(http://www.xmlspy.com)by
                                                                    Myself(4D SA)-->
<?PI TextProcess?>
<!DOCTYPE RootElement SYSTEM"ParseTest.dtd">
```

次の命令を実行すると

SAX GET XML PROCESSING INSTRUCTION(\$RefDoc;vName;vValue)

vName には “PI” が、vValue には “TextProcess” が返されます。

SAX GET XML ELEMENT

SAX GET XML ELEMENT(文書; 要素名; プレフィックス; 属性名; 属性値)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
要素名	文字列	← 要素の名前
プレフィックス	文字列	← 名前空間
属性名	文字列配列	← 属性の名前
属性値	文字列配列	← 属性の値

このコマンドは、引数<文書>で参照される XML 文書内に存在する要素<要素名>に関する各種情報を返します。このコマンドは、「XML Start Element」イベントまたは「XML End Element」イベントとともに呼び出さなくてはなりません。「XML End Element」イベントの特例として、属性パラメータが処理されません。

引数<要素名>には要素の名前を格納されます。

引数<プレフィックス>には要素の名前空間（ネームスペース）が代入されます。要素に名前空間が関連付けられていない場合、この引数は空の文字列を返します。

コマンドは、配列<属性名>に対象となる要素の属性名を代入します。必要に応じて、コマンドは配列を自動的に作成し、そのサイズを調整します。

また、コマンドは、配列<属性値>に対象となる要素の属性の値を代入します。必要に応じて、コマンドは配列を自動的に作成し、そのサイズを調整します。

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 次のXMLコードについてみてみましょう。

```
<RootElement>
<Child Att1="111"Att2="222"Att3="333">MyText</Child>
</RootElement>
```

次の命令を実行すると

SAX GET XML ELEMENT(DocRef;vName;vPrefix;attrNames;attrValues)

vNameには“Child”

vPrefixには“”

attrNames{1}には“Att1”、attrNames{2}は“Att2”、attrNames{3}は“Att3”

attrValues{1}には“111”、attrValues{2}には“222”、attrValues{3}には“333”

がそれぞれ代入されます。

SAX GET XML ELEMENT VALUE

SAX GET XML ELEMENT VALUE(文書;値)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
値	テキスト BLOB	← 要素の値

SAX GET XML ELEMENT VALUE コマンドを使用し、引数<文書>で参照されるXML文書内に存在するXML要素の<値>を取得することができます。このコマンドは、「**SAX XML DATA**」イベントとともに呼び出さなくてはなりません。

引数<値>には、テキストまたはBLOBタイプの変数を渡します。BLOBを渡した場合、テキストはそのままの状態で見返されます（修正されない）。

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 次のXMLコードについてみてみましょう。

```
<RootElement>
<Child Att1="111"Att2="222"Att3="333">MyText</Child>
</RootElement>
```

次の命令を実行すると

```
SAX GET ELEMENT VALUE(DocRef;vValue)
```

vValueには“MyText”が返されます。

SAX GET XML CDATA

SAX GET XML CDATA(文書;値)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
値	BLOB ←	要素の値

SAX GET XML CDATA コマンドを使用し、引数<文書>で参照されるXML文書内に存在するXML要素のCDATAの<値>を取得することができます。このコマンドは、「SAX XML CDATA」イベントとともに呼び出さなくてはなりません。

データはそのままの状態で見返されます（データは修正されない）。

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 次のXMLコードについてみてみましょう。

```
<RootElement>
<Child>MyText<![CDATA[MyCData</Child>
</RootElement>
```

次の4Dコードを実行した場合

```
C_BLOB(vData)
C_TEXT(vTextData)
SAX GET XML CDATA(DocRef;vData)
vTextData:=BLOB to text(vData;C string)
```

vTextDataには“MyCData”が返されます。

SAX GET XML ENTITY

SAX GET XML ENTITY(文書; 名前; 値)

引数	タイプ	説明
文書	文書参照	→ 開かれている文書の参照番号
名前	文字列	← エンティティ (実体) の名前
値	文字列	← エンティティの値

SAX GET XML ENTITY コマンドを使用し、引数<文書>で参照されるXML文書内に存在するXMLエンティティの<値>を取得することができます。このコマンドは、「SAX XML Entity」イベントとともに呼び出さなくてはなりません。

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 次のXMLコードについてみてみましょう。

```
<?xml version="1.0"encoding="UTF-8"?>
<!DOCTYPE body[
  <ELEMENT body(element*)>
  <ELEMENT element(#PCDATA)>
  <ENTITY name"Replacement">
]>
<body>
  <element>Entity updated by&name;</element>
</body>
```

次の命令を実行すると

SAX GET XML ENTITY(DocRef;vName;vValue)

vNameには“name”が、vValueには“Replacement”が返されます。

DOM コマンドを用いてXMLを設定する

4th Dimension 2004では、一連のDOMコマンドが新たに提供され、XMLツリーをその標準に合わせて設定または編集することができます。

注：DOM規格と、それに代わる方法であるSAX標準に関する詳細は、前述の「DOMおよびSAXとは？」の節を参照してください。

XPath ノーテーションの管理

新しい2つのDOMコマンド (**DOM Create XML element** および **DOM Get XML element**) は、XML 要素へアクセスするためのXPathノーテーションに対応します。XPathノーテーションはXPath言語に由来し、XML構造内をナビゲーションすることを目的としています。XPathノーテーションにより、フルアクセスパスを指示しなくても、XML構造内でダイレクトに要素を設定することができます。

次の構造は、この例を表わしています。

```
<RootElement>
  <Elem1>
    <Elem2>
      <Elem3 Font=Verdana Size=10></Elem3>
    </Elem2>
  </Elem1>
</RootElement>
```

XPathノーテーションを使用すると、シンタックス “RootElement/Elem1/Elem2/Elem3” を用いて要素3にアクセスすることができます。

4 th Dimension 2004では、シンタックス “要素[要素番号]” を用いて、指標付けされたXPath要素にも対応します。

次の構造は、この例を表わします。

```
<RootElement>
  <Elem1>
    <Elem2>aaa</Elem2>
    <Elem2>bbb</Elem2>
    <Elem2>ccc</Elem2>
  </Elem1>
</RootElement>
```

XPathノーテーションにより、シンタックス “RootElement/Elem1/Elem2[3]” を用いて値 “ccc” にアクセスすることができます。

注：依然として4th Dimension 2004では、DOM Create XML element および DOM Get XML element コマンドに指定された引数 <xPath> は、引数として渡されたXML要素参照に関連するものとみなされます。従って、この値は “/” 文字で始まらない可能性があります (この文字は考慮されない)。

XPathノーテーションの説明は、**DOM Create XML element** および **DOM Get XML element** コマンドの例題を参照してください。

DOM SET XML OPTIONS

DOM SET XML OPTIONS(要素参照;エンコード{; スタンドアロン{; インデント})

引数	タイプ	説明
要素参照	文字列	→ XML 要素参照
エンコード	文字列	→ XML 文書の文字セット
スタンドアロン	ブール	→ True= 文書はスタンドアロン False (デフォルト) = 文書はスタンドアロンではない
インデント	ブール	→ True (デフォルト) = 文書はインデントされる False= 文書はインデントされない

DOM SET XML OPTIONS コマンドにより、<要素参照>を用いて指定されたXMLツリーの作成に役立つ各種オプションを定義することができます。これらのオプションは、符号化方式、スタンドアロン属性、ならびにツリーのインデントに関連します。

- 符号化方式：文書で使用する文字セットを示します。デフォルトでは（このコマンドが呼び出されない場合）、UTF-8文字セット（圧縮Unicode）が使用されます。
- スタンドアロン：ツリーが独立しているか（True）、または動作するために他のファイルや外部リソースが必要となるか（False）を示します。デフォルトでは（このコマンドが呼び出されない場合や引数が省略された場合）、ツリーは独立していません。
- インデント：XMLキーの階層構造に従ってツリーのインデントを表示するか（True）、表示しないか（False）を示します。デフォルトでは（このコマンドが呼び出されない場合や引数が省略された場合）、ツリーはインデントされます。

▼ 次の例題は、要素<要素参照>で使用する符号化方式とスタンドアロンオプションを設定します。

```
DOM SET XML OPTIONS(DocRef;"UTF16";True)
```

参照

DOM Create XML Ref

DOM Create XML Ref

DOM Create XML ref(ルート{; 名前空間{; 名前空間名; 名前空間値} {; 名前空間名 X; 名前空間値 X}) → 文字列

引数	タイプ		説明
ルート	文字列	→	ルート要素の名前
名前空間	文字列	→	名前空間の値
名前空間名	文字列	→	名前空間の名前
名前空間値	文字列	→	名前空間の値
戻り値	文字列	←	ルート XML 要素の参照

このコマンドは、メモリ上に空のXMLツリーを作成し、その参照を返します。

引数<ルート>には、XMLツリーのルート要素の名前を渡します。

任意の引数<名前空間>には、ツリーの名前空間の値の宣言を渡します（例：“http:www.4d.com”）。

その際、引数<ルート>の前には、名前空間の値に続けてコロン(:)を付加しなくてはなりません（例：“MyNamespace:MyRoot”）。

注：<名前空間>は文字列であり、この値によりXML変数名が確実に一意となるように設定することができます。通常、http://www.mysite.com/myurlのようなURLが用いられます。このURLは必ずしも有効でなくても構いませんが、一意（ユニーク）でなければなりません。詳細については、4th Dimensionの『デザインリファレンス』を参照してください。

<名前空間名>と<名前空間値>のペアを使用して、生成されるXMLツリー内に1つ以上の名前空間をさらに宣言することができます。名前空間の名前/値のペアを必要な数だけ渡すことができます。

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 1つのXMLツリーを作成します。

```
C_STRING(16;vElemRef)
vElemRef:=DOM Create XML reference("MyRoot")
```

このメソッドの結果は次の通りです。

```
<?xml version="1.0"encoding="UTF-8"standalone="no"?>
<MyRoot/>
```

▼ 1つの名前空間があるXMLツリーを作成します。

```
C_STRING(16;vElemRef)
$Root:="MyNameSpace:MyRoot"
$Namespace:="http://www.4D.com/tech/namespace"
vElemRef:=DOM Create XML reference($Root;$Namespace)
```

このメソッドの結果は次の通りです。

```
<MyNameSpace MyRoot xmlns:MyNameSpace"http://www.4D.com/tech/namespace"/>
```

▼ 複数の名前空間があるXMLツリーを作成します。

```
C_STRING(16;vElemRef)
C_STRING(80;$aNSName1;$aNSName2;$aNSValue1;$aNSValue2)
$Root:="MyNameSpace:MyRoot"
$Namespace:="http://www.4D.com/tech/namespace"
aNSName1:="NSName1"
aNSName2:= "NSName2"
aNSValue1:="http://www.4D.com/Prod/namespace"
aNSValue2:="http://www.4D.com/Mkt/namespace"
vElemRef:=DOM Create XML reference
($Root;$aNSName1;$aNSValue1;aNSname2;aNSValue2)
```

このメソッドの結果は次の通りです。

```
<MyNameSpace MyRoot xmlns:MyNameSpace"http://www.4D.com/tech/nameSpace"
NSName1"http://www.4D.com/Prod/namespace"
NSName2"http://www.4D.com/Mkt/namespace"/>
```

DOM SET XML ATTRIBUTE

DOM SET XML ATTRIBUTE(要素参照; 属性名; 属性値...{; 属性名 X; 属性値 X})

引数	タイプ	説明
要素参照	文字列	→ XML 要素参照
属性名	文字列	→ セットする属性
属性値	文字列	→ 属性の新しい値

このコマンドを使用すると、引数<要素参照>に渡した参照番号を持つXML要素に対し、1つ以上の属性を追加することができます。また、定義される各属性の値も設定することができます。

設定する属性ならびにその値は、それぞれ引数<属性名>と<属性値>に渡します（変数、フィールド、リテラル値の形で）。必要な数の属性と値のペアを渡すことができます。

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、それ以外の場合は0が代入されてエラーが返されます。

▼ 次のXMLソースの場合

```
<Book>
<Title>The Best Seller</Title>
</Book>
```

次のコードを実行します。

```
vAttrName:="Font"
vAttrVal:="Verdana"
DOM SET XML ATTRIBUTE(vElemRef;vAttrName;vAttrVal)
```

この結果は次の通りです。

```
<Book>
<Title Font=Verdana>The Best Seller</Title>
</Book>
```

DOM Create XML element

DOM Create XML element(要素参照; XPath{; 属性名; 属性値}...{; 属性名 X; 属性値 X}) → 文字列

引数	タイプ	説明
要素参照	文字列	→ ルートXML要素の参照
xPath	テキスト	→ 作成するXML要素のXPathパス
属性名	文字列	→ 設定する属性
属性値	文字列	→ 属性の新しい値
戻り値	文字列	← 作成したXML要素の参照

この関数を使用し、引数<XPath>に指定したパスのXML要素<要素参照>内に新しい要素を作成することができます。

<要素参照>には、ルート要素の参照を渡してください（例えば、**DOM Create XML element** コマンドで作成したルート要素）。

<XPath>には、作成する要素のアクセスパスをXMLフォーマットで渡します（前述の「XPathノテーションの管理」の節を参照）。そのパスに要素が存在しない場合は、要素が作成されます。

注：要素参照で指定したツリーに対し、1つ以上の名前空間を定義している場合（前述のDOM Create XML Ref関数を参照）、使用する名前空間名を引数<XPath>の前におかなければなりません（例：“MyNameSpace:MyElement”）。

任意の引数<属性名>と<属性値>には、属性と値のペアを渡すことができます（変数、フィールド、リテラル値の形で）。必要な数のペアを渡すことができます。

コマンドは結果として作成された要素のXML参照を返します。

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、それ以外の場合は0が代入されてエラーが返されます。エラーが返される条件は次の通りです。

■ ルート要素の参照が無効。

■ 作成する要素の名前が無効（例えば、数値で始まる名前）。

▼ 次のような要素を作成してみましょう。

```
<?xml version="1.0"encoding="UTF-8"standalone="no"?>
<RootElement>
  <Elem1>
    <Elem2>
      <Elem3></Elem3>
    </Elem2>
  </Elem1>
</RootElement>
```

これを作成するには、次のように記述します。

```
C_STRING(16;vRootRef;vRefElement)
vRootRef:=DOM Create XML Ref("RootElement")
vxPath:="/Elem1/Elem2/Elem3"
vRefElement:=DOM Create XML element(vRootRef;vxPath)
```

▼ 次のような要素を作成してみましょう（属性を含む）。

```
<?xml version="1.0"encoding="UTF-8"standalone="no"?>
<RootElement>
  <Elem1>
    <Elem2>
      <Elem3 Font=Verdana Size=10></Elem3>
    </Elem2>
  </Elem1>
</RootElement>
```

これを作成するには、次のように記述します。

```
C_STRING(16;vRootRef;vElemRef)
C_STRING(80;$AttrName1;$AttrName2;$AttrVal1;$AttrVal2)
$AttrName1:="Font"
```

```
$AttrName2:="Size"  
$AttrValue1:="Verdana"  
$AttrValue2:="10"  
vRootRef:=DOM Create XML Ref("RootElement")  
vxPath:="/Elem1/Elem2/Elem3"  
vElemRef:=DOM Create XML element  
          (vRootRef;vxPath;$AttrName1;$AttrValue1;$AttrName2;$AttrVal2)
```

DOM SET XML ELEMENT NAME

DOM SET XML ELEMENT NAME(要素参照; 要素名)

引数	タイプ	説明
要素参照	文字列	→ XML 要素の参照
要素名	文字列	→ 要素の新しい名前

このコマンドを使用し、<要素参照>で指定した要素の名前を変更することができます。

<要素参照>には名前を変更する要素の参照を渡し、<要素名>には要素の新しい名前を渡します。さらに、このコマンドは要素の開始タグと終了タグの更新も行います。

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、それ以外の場合は0が代入されてエラーが返されます。エラーが返される条件は次の通りです。

- 要素参照が無効。
- 作成する要素の新しい名前が無効（例えば、数値で始まる名前）。

▼ 次のようなXMLソースの場合

```
<Book>  
  <Title>The Best Seller</Title>  
</Book>
```

要素“Book”への参照をvElemRefに代入し、次のコードを実行します。

```
DOM SET XML ELEMENT NAME(vElemRef;"BestSeller")
```

結果は次の通りです。

```
<BestSeller>  
  <Title>The Best Seller</Title>  
</BestSeller>
```

DOM SET XML ELEMENT VALUE

DOM SET XML ELEMENT VALUE(要素参照; 要素値)

引数	タイプ	説明
要素参照	文字列	→ XML 要素の参照
要素値	文字列; 変数	→ 要素の新しい値

このコマンドを使用し、<要素参照>で指定した要素の値を変更することができます。

<要素参照>には、指定した要素の新しい値を格納した文字列、または変数（またはフィールド）を渡します。

■ 文字列を渡すと、その値は“現状のまま”XML構造上で使用されます。

■ 変数またはフィールドを渡すと、4th Dimension は<要素値>のタイプに応じてその値を処理します。配列、ピクチャとポインタ以外のすべてのデータタイプを使用することができます。

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、それ以外の場合は0が代入されてエラーが返されます（例えば、要素参照が無効）。

▼ 次のようなXMLソースの場合

```
<Book>
  <Title>The Best Seller</Title>
</Book>
```

要素“Title”への参照をvElemRefに代入し、次のコードを実行します。

```
DOM SET XML ELEMENT VALUE(vElemRef;"The Loser")
```

結果は次の通りです。

```
<Book>
  <Title>The Loser</Title>
</Book>
```

DOM REMOVE XML ELEMENT

DOM REMOVE XML ELEMENT(要素参照)

引数	タイプ	説明
要素参照	文字列	→ XML 要素の参照

このコマンドをは、<要素参照>で指定した要素を削除します。

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、それ以外の場合は0が代入されてエラーが返されます。エラーが返される条件は次の通りです。

■ 要素参照が正しくない

■ 要素が空

DOM EXPORT TO FILE

DOM EXPORT TO FILE(要素参照; ファイルパス)

引数	タイプ	説明
要素参照	文字列	→ ルートXML要素の参照
ファイルパス	テキスト	→ ファイルのフルアクセスパス

このコマンドを使用し、XMLツリーをディスク上のファイルに保存することができます。

引数<要素参照>には、書き出すルート要素の参照を渡します。

<ファイルパス>には、使用または作成する書き出しファイルのフルアクセスパスを渡します。ファイルが存在しない場合は作成されます。

ファイル名だけを渡した場合（アクセスパスなし）、そのファイルの検索が行われるか、またはストラクチャファイルと同じ階層にファイルが作成されます。

空の文字列("")を渡すと、ファイルや新規ファイルを開く標準のダイアログボックスが表示されます。

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、それ以外の場合は0が代入されてエラーが返されます。エラーが返される条件は次の通りです。

■ 要素参照が無効

■ 指定したアクセスパスが無効

■ 保存先のボリュームからエラーが返される（ディスクの空き容量が不十分など）

この例題は、ツリー「vElemRef」を「MyDoc.xml」ファイルに保存します。

DOM EXPORT TO FILE(vElemRef;"C:¥¥folder¥¥MyDoc.xml")

DOM EXPORT TO VAR

DOM EXPORT TO VAR(要素参照; vXmlVar)

引数	タイプ	説明
要素参照	文字列	→ ルートXML要素の参照
vXmlVar	テキスト BLOB	→ XMLツリーを受け取る変数

このコマンドを使用し、XMLツリーをテキストまたはBLOBタイプの変数に保存することができます。

引数<要素参照>には、書き出すルート要素の参照を渡します。

<vXmlVar>には、XMLツリーを格納する変数の名前を渡します。この変数は、テキストまたはBLOBのいずれかのタイプでなくてはなりません。次に実行しようとする動作や、ツリーの予想サイズ（テキストタイプの変数のサイズは、32,000桁までに制限される点に注意）に応じて、そのタイプを選択することができます。

要素がテキスト変数に保存される場合、「Mac Roman」符号化方式が使用されます。

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、それ以外の場合は0が代入されてエラーが返されます（例えば、要素参照が無効）。

▼この例題は、ツリー「vElemRef」をテキスト変数に保存します。

```
C_TEXT(vtMyText)
DOM EXPORT TO VAR(vElemRef;vtMyText)
```

DOM Find XML element

DOM Find XML element(要素参照; XPath) → 文字列

引数	タイプ	説明
要素参照	文字列	→ XML要素の参照
xPath	テキスト/BLOB	→ 検索する要素のXPathパス
戻り値	要素参照	← 発見された要素の参照（該当する場合）

このコマンドを使用し、XML構造内において特定のXML要素を検索することができます。検索は、引数<要素参照>で指定した要素から開始されます。

検索するXMLノードは、引数<xPath>を用いて指定します。XPathフォーマットを使用することができます（前述の「XPathノーテーションの管理」の節を参照）。指標付きの要素も使用可能です。

注：XML標準に準拠して、検索では大文字小文字が区別されます。

コマンドは、発見された要素のXML参照を返します。

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、それ以外の場合は0が代入されてエラーが返されます。エラーが返される条件は次の通りです。

■ 要素参照が無効

■ 指定したXPathパスが無効

▼ この例題により、XML要素を迅速に検索し、その値を表示することができます。

```
vFound:=DOM Find XML element(vElemRef;"/Book[15]/Title")
DOM Get XML ELEMENT VALUE(vFound;value)
ALERT("要素の値： ¥"+value+"¥")
```

▼ 同様の検索は、次の式を用いて実行することもできます。

```
vFound:=DOM Find XML element(vElemRef;"/Book[15]")
vFound:=DOM Find XML element(vFound;"/Title")
DOM GET XML ELEMENT VALUE(vFound;value)
ALERT("要素の値： ¥"+value+"¥")
```

変更されたコマンド

DOM GET XML ELEMENT VALUE

DOM GET XML ELEMENT VALUE(要素参照; 要素値{; cDATA})

引数	タイプ	説明
要素参照	文字列	→ XML 要素参照
要素値	変数	← 要素の値
cDATA 変数		← CDATA セクションの内容

DOM GET XML ELEMENT VALUE コマンドは、追加された任意の引数<CDATA>を受け入れます。この引数を使用して、XML要素<要素参照>のCDATAセクションの内容を受け取ります。

引数<要素値>の場合と同様に、4th Dimensionは取得した値を変換し、引数として渡した変数と同じタイプに変えようとしています。

XML エラーコード

XMLに関連する4th Dimensionのエラーコードを次に示します。

SAXエラー

- 9915 文書の参照が無効です。
- 9916 要素が開かれていません。
- 9917 引数に渡された配列タイプが無効です。
- 9918 要素の名前が無効です。
- 9919 この符号化方式はサポートされません。
- 9920 ノードのタイプが無効です。

DOMエラー

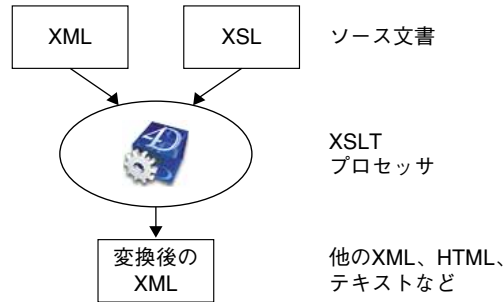
- 9925 参照された要素がヌルです。
- 9926 参照された要素が無効です。
- 9927 参照された要素は“ルート”ではありません。
- 9928 未知の要素名です。
- 9929 この要素の指標が無効です。
- 9930 この要素には、その名前の属性が存在しません。
- 9931 この属性の指標が無効です。
- 9932 XML DLL がロードされていません。
- 9933 XML ファイルが無効です。
- 9934 XML ファイルが整形形式ではありません。
- 9935 XML ファイルが無効、または整形形式ではありません。

XSL Transformations の管理

4th Dimension 2004 は、XSL スタイルシートのアプリケーション (eXtended Stylesheet Language) をサポートします。XSL 言語を使用し、XML 文書のタグを変更することができます。

XSL 言語には2種類の機能があります。

- **Formatting (フォーマット)** : この機能により、HTML の CSS (Cascading Style Sheets) と同様に、XML 要素に対してスタイルや表示規則を適用することができます。
- **Transformations (トランスフォーメーション)** : この機能により、XML タグを別のタグシステムに変換することができます (例えば、HTML)。この機能は、XSLT として知られています。XSL スタイルシートは、文書内の XML 要素を選択することにより、要素全体を再編成し、別の要素に変換します。例えば、この機能は一連の異なる文書を同期化する場合に役立ちます。



注：4th Dimension は、XSL 変換を実行するために「Xalan-C_1_6_0.dll」ライブラリを利用します。Xalan はフリーウェアの XSL プロセッサです。詳細については、<http://xml.apache.org/xalan-c/index.html> を参考にしてください。

XSL スタイルシートはテキストタイプの文書で（拡張子 “.xsl” が付属）、手動または専用のアプリケーションを用いて生成されます。その全体的な構造は CSS とほぼ同じです。XSL 言語にはさまざまな要素や機能が用意されており、あらゆるタイプのダイナミック変換を実行することができます。

この言語に関する詳細は、<http://xmlfr.org>（一例として）を参考にしてください。

4th Dimension 2004 では、既存の XSL スタイルシートを用いて XML 文書を変換することができます（**APPLY XSLT TRANSFORMATION** を参照）。また、4th Dimension では **SET XSLT PARAMETER** コマンドを使用して、XSL スタイルシートのパラメータをオンザフライで変更することも可能です。

注：書き出し用のダイアログボックスに追加された新しいオプションを使用すると、XML 書き出し時に XSL スタイルシートを利用することが可能であり、その結果、変換済みの XML 文書が生成されます（前述の「XSL 変換を適用する」の節を参照）。

APPLY XSLT TRANSFORMATION

APPLY XSLT TRANSFORMATION(xml ソース; xsl シート; 結果)

引数	タイプ	説明
xml ソース	文字列; BLOB	→ XML ソース文書の名前またはアクセスパス、または XML ソースを納めた BLOB
xsl シート	文字列; BLOB	→ XSL スタイルシートを納めた文書の名前またはアクセスパス、または XSL スタイルシートを納めた BLOB
結果	文字列; BLOB	→ XSLT 変換の結果を受け取る文書の名前またはアクセスパス、または XSLT 変換の結果を受け取る BLOB

APPLY XSLT TRANSFORMATION コマンドは、XML を納めた文書や BLOB に対して XSLT 変換を適用し、文書または BLOB を生成します。

コマンドは、3つの BLOB または文字列タイプの引数を必要とします。

警告：このコマンドは、変数またはフィールドのみを引数として受け入れます。

文字列を渡した場合は、文書を指定します。その際、文書の名前（文書はデータベースストラクチャと同じ階層に置かなくてはならない）またはフルアクセスパスだけを渡すことができます。

一度の呼び出しにおいて、異なるタイプの引数を混ぜて使用することができます。しかし、空の引数を渡すことはできません（BLOB またはテキスト）。

■ 引数<xml ソース>には、変換する XML ソースを納めなくてはなりません。コマンドは XML コードの妥当性を検証しないため、XML コードは必ず妥当でなくてはなりません。

■ <xsl シート>には、XSLT 変換に使用する XSL スタイルシートを納めます。このスタイルシートは、手動または専門のソフトウェアを用いて生成されます。コマンドは XSL シンタクスの妥当性を検証しないため、XSL シンタクスは必ず妥当でなくてはなりません。

■ 引数<結果>には、XSLT 変換の結果を受け取る文書の名前または BLOB を渡さなくてはなりません。渡した文書名が指定した場所に存在しない場合、4th Dimension はその文書を自動的に作成します。文書が書き込みアクセスで既にオープンされている場合、エラーが生成されます。

コマンドは XML ソースを解析し、それを XSL スタイルシートの命令を用いて変換します。事前に **SET XSLT PARAMETER** コマンドが使用されている場合、コマンドはその値により指定された引数を置き換えます。変換結果は文書または BLOB の<結果>に書き込まれます。

変換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、それ以外の場合は 0 が代入されます。

注：このコマンドのパフォーマンスを最適化するため、特に同じ XSL スタイルシートから繰り返し使用されるアプリケーションにおいて、4th Dimension は前回使用された XSL スタイルシートをコンパイルし、それをメモリ上に保存しておきます。

▼ **SET XSLT PARAMETER** コマンドの例題を参照してください。

参照

SET XSLT PARAMETER

SET XSLT PARAMETER

SET XSLT PARAMETER(パラメータ名; パラメータ値)

引数	タイプ	説明
パラメータ名	文字列	→ XSLシート内で検索するパラメータの名前
パラメータ値	文字列	→ 変換後の文書において使用するパラメータの値

SET XSLT PARAMETER コマンドは、必ず **APPLY XSLT TRANSFORMATION** コマンドと一緒に使用しなくてはなりません。これにより、XML文書のXSLT変換を開始する際に、XSLスタイルシート内にある変数のパラメータの値を設定することができます。

このコマンドを使用すると、**APPLY XSLT TRANSFORMATION** コマンドの使用直前に、4th Dimension プロセスからもたらされた値をXSLスタイルシートに挿入することができます。

<パラメータ名>には、置き換えるXSL変数のパラメータ名を渡します。XSLスタイルシートにおいて、このパラメータは“Storeplace”のような形で存在していません。しかし、<パラメータ名>に“\$”記号は不要です。例えば、<xsl:template match=\$myvar>という命令をXSLファイルに指定する場合は、<パラメータ名>に“myvar”だけを渡し、このパラメータを指定します。

変換後のファイルに挿入したい値は、<パラメータ値>で指定したXSL変数に格納します。前述の例題を用いると、<パラメータ値>に“title”を渡した場合は、XSLT変換において<xsl:template match="title">という命令が考慮されます（これは要素“title”を対象としてスタイル規則に設定）。

注：XSL言語の詳しい説明に関しては、この言語専用につけられたインターネット上の数々のサイトを参照することができます（例：<http://xmlfr.org>）。

XSLスタイルシートの複数のパラメータを渡すには、**APPLY XSLT TRANSFORMATION** コマンドを複数呼び出します。同一プロセス内で**APPLY XSLT TRANSFORMATION** が呼び出されるまで、各パラメータが“積み重ね”られます。**APPLY XSLT TRANSFORMATION** コマンドが実行されると、“積み重ね”られたパラメータは自動的に削除されます。

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、それ以外の場合は0が代入されます。

▼ 次の例題は、2つのXSLパラメータを設定した後、スタイルシート「mysheet.xml」を用いて文書「mydoc.xml」をHTMLファイルに変換します。

```
SET XSLT PARAMETER("varstyle";"bold")
```

```
SET XSLT PARAMETER("varcolor";"blue")
$xmlidoc:="mydoc.xml"
$xslsheet:="mysheet.xsl"
$htmlidoc:="mydoc.html"
APPLY XSLT TRANSFORMATION($xmlidoc;$xslsheet;$htmlidoc)
```

参照

APPLY XSLT TRANSFORMATION

GET XSLT ERROR

GET XSLT ERROR(エラーテキスト{; 行{; 桁位置})

引数	タイプ		説明
エラーテキスト	変数	←	エラーのテキスト
行	変数	←	行番号
桁位置	変数	←	桁位置

このコマンドは、カレントプロセスにおいて実行したXSLT変換中に発見された最後のエラーの説明を引数<エラーテキスト>に代入して返します。返される情報は、「Xerces.DLL」ライブラリから提供されています。

任意の引数<行>と<桁位置>は、XSL文書内のエラー箇所を示します。これらの引数にはそれぞれ、行番号およびその行の中でエラー原因となった式の最初の文字位置が代入されます。

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。エラーが発生すると0が代入されます。

Web サーバ (クライアント側)

SET WEB SERVICE OPTION

SET WEB SERVICE OPTION(オプション; 値)

引数	タイプ		説明
オプション	倍長整数	→	設定するオプションのコード
値	倍長整数	→	オプションの値

注：このコマンドは上級Webサービスユーザを対象としています。このコマンドの使用は任意です。

SET WEB SERVICE OPTION コマンドを使用すると、**CALL WEB SERVICE** コマンドで次の SOAP リクエストが開始された時に使用する各種オプションを設定することができます。

CALL WEB SERVICE コマンドの前に、設定したいオプションの数だけ、このコマンドを呼び出すことができます。

引数<オプション>には、設定するオプションの値を引数<値>に代入して渡します。

引数<オプション>に対し、事前に定義された「Web Services(Client)」テーマの定数のいずれかを使用することができます。

■ Web Service SOAP Header

■ Web Service HTTP Timeout

■ Web Service SOAP Version

次の節では、これらの利用可能なオプションと値についてそれぞれ説明します。

■ Web Service SOAP Header

<値>=SOAP リクエストのヘッダとして挿入する XML ルート要素の参照。

このオプションにより、**CALL WEB SERVICE** コマンドを用いて生成された SOAP リクエストにヘッダを挿入することができます。デフォルトでは、SOAP リクエストに特定のヘッダは含まれていません。しかし、`identification` パラメータの管理のような一定の Web サービスでは、ヘッダが必要になります。

■ Web Service HTTP Timeout

<値>=クライアント側のタイムアウト (秒単位)

クライアント側のタイムアウトとは、サーバからの応答がない場合の Web サービスクライアントの待機時間です。この待機時間が過ぎると、クライアントはセッションをクローズし、リクエストは失われます。

デフォルトのタイムアウトは 10 秒間です。特別な理由から (ネットワークステータス、Web サービスの特殊事情など)、この値を変更することができます。

■ Web Service SOAP Version

<値>=Web Service SOAP_1_1 または Web Service SOAP_1_2 (「Web Services(Client)」テーマの定数)

このオプションにより、リクエストで使用する SOAP プロトコルのバージョンを指定することができます。バージョン 1.1 を指定するには、<値>に `Web Service SOAP_1_1` を渡し、バージョン 1.2 を指定するには <値>に定数 `Web Service SOAP_1_2` を渡します。

このオプションの呼び出し順序は重要ではありません。同じくオプション>が何度も設定された場合、最後の呼び出しの値が考慮されます。

▼ 次の例題は、独自のヘッダを SOAP リクエストに挿入します。

```
`XML 参照の作成
C_STRING(16;vRootRef;vElemRef)
vRootRef:=DOM Create XML element("RootElement")
vxPath:="/Elem1/Elem2/Elem3"
vElemRef:=DOM Create XML element(vRootRef;vxPath)
`参照を用いて SOAP ヘッダを変更する
SET WEB SERVICE OPTION(Web Service SOAP Header;vElemRef)
```

▼ 次の例題は、SOAP プロトコルバージョン 1.2 を使用します。

```
SET WEB SERVICE OPTION(Web Service SOAP Version; Web Service SOAP_1_2)
```

外部データソース

この新しいテーマに納められたコマンドを使用すると、標準のプロトコルを用いて 4th Dimension から他のアプリケーションに保存されているデータへアクセスすることができます。4th Dimension 2004 では、ODBC コマンドだけが利用可能です。

ODBC (Open DataBase Connectivity) 規格は、標準的な機能のライブラリを定義します。これらの機能を使用すると、4th Dimension のようなアプリケーションが SQL を用いて ODBC 互換データソース (データベース、スプレッドシートなど) へアクセスすることができます。

注：4th Dimension 2004 では、「ユーザ」モードにおいて ODBC ソースに対するデータの書き出しと読み込みを行うことができます (前述の「ODBC データソースを使用した読み込みと書き出し」の節を参照)。

4th Dimension の「外部データソース」テーマに組み込まれたハイレベル ODBC コマンドを使用すると、お使いの 4th Dimension アプリケーションと ODBC データソースが交信を行えるシンプルなソリューションを実装することができます。アプリケーションにおいて、より高度な ODBC 機能が必要となる場合は、4D から提供される“ローレベル” ODBC プラグインを入手してください (後述の注意を参照)

4D ODBC ユーザへの注意：現行の 4D ODBC プラグインは、4th Dimension 2004 との互換性がありません。4D からは次世代の ODBC プラグインが提供されており、これを用いて複雑な ODBC 接続を管理することができます。また、特定のプラグインも 4D から提供され、これによりデータベースにある従来の 4D ODBC プラグインコマンドを新しいプラグインコマンドへ変換することができます。詳細については、4th Dimension 2004 において提供される関連のドキュメントを参照してください。

4th Dimension 2004でODBCコマンドを使用する上で役立つ情報を次に示します。

■ ログインの有効範囲はプロセスです。同時接続を管理したい場合は、それぞれのODBC LOGINに対して1つの4Dプロセスを開始しなくてはなりません。

ODBC CANCEL LOAD コマンドを使用して、同一接続における複数の**SELECT**リクエストを実行します。

■ すべてのODBCコマンドの後に**ON ERR CALL** コマンドを使用すると、返されるODBCエラーを捕捉することができます。その際、**ODBC GET LAST ERROR** コマンドは追加情報を取得するために使用されます。

ODBC LOGIN

ODBC LOGIN({データエントリ}; ユーザ名; パスワード}}

引数	タイプ	説明
データエントリ	文字列	→ ODBCマネージャにおけるデータソースエントリの名前
ユーザ名	文字列	→ データソースに登録されたユーザの名前
パスワード	文字列	→ データソースに登録されたユーザのパスワード

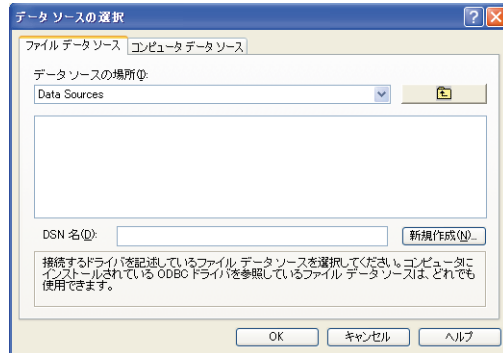
ODBC LOGIN コマンドを使用すると、外部のODBCデータソースへ接続することができます。

引数<データエントリ>には、ODBCドライバマネージャにおいて入力されたデータソースの名前を納めます。

<ユーザ名>には、外部データソースへの接続を許可されたユーザの名前を納めます。例えば、Oracleを使用している場合のユーザ名は“Scott”です。

<パスワード>には、外部データソースへの接続を許可されたユーザのパスワードを納めます。例えば、Oracleを使用している場合のパスワードは“tiger”です。

これらの引数はオプションです。引数をまったく渡さない場合は、コマンドにより「ODBCマネージャ」ダイアログボックスが表示され、外部データソースを選択することができます。



このコマンドの有効範囲は、各プロセスです。つまり、異なる2つの接続を行いたい場合、2つのプロセスを作成してそれぞれのプロセス内で各接続を実行しなくてはなりません。

接続が正常に終了すると、システム変数OKには1が代入され、それ以外の場合は0が代入されます。

▼ この命令は、「ODBC ドライバ」マネージャウインドウを表示します。

ODBC LOGIN

▼ この命令は、名前とパスワードに“Scott”と“tiger”を使用して、“MyOracle”という名前の外部データソースへ接続します。

ODBC LOGIN("MyOracle";"Scott";"tiger")

ODBC LOGOUT

ODBC LOGOUT

引数	タイプ	説明
----	-----	----

このコマンドに引数は必要ありません。

ODBC LOGOUT コマンドは、カレントプロセスの接続をクローズします (該当する場合)。接続が行われていない場合、コマンドは何も行いません。

ログアウト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、それ以外の場合は0が代入されます。

ON ERR CALL コマンドにより設定されたエラー処理用メソッドを用いて、このエラーを遮ることができます。

ODBC SET OPTION

ODBC SET OPTION (オプション; 値)

引数	タイプ	説明
オプション	倍長整数	→ 設定するオプションの番号
値	倍長整数	→ オプションの新しい値

ODBC SET OPTION コマンドを使用して、<オプション>に渡したオプションの<値>を変更します。

<オプション>には、「External Data Source」テーマにある次の定数のうちいずれかを納めることができます。

定数	説明とその値
ODBC Asynchronous(1)	0= 同期接続 (デフォルト値) 1 (または0以外の値) =非同期接続
ODBC Max Rows(2)	結果のグループにおける最大行数 (プレビューで使用)
ODBC Max Data Length(3)	返されるデータの最大長
ODBC Query Timeout(4)	ODBC EXECUTE コマンドの実行時に応答を待機する最大タイムアウト この値が考慮されるためには、接続をオープンする前にこの値を設定しなくてはなりません。 値: 時間 (秒単位) 省略値: タイムアウトなし
ODBC Connection Timeout(5)	ODBC LOGIN コマンドの実行時の最大タイムアウト 可能な値: 時間 (秒単位) 省略値: タイムアウトなし コマンドが正常に終了すると、システム変数 OK には 1 が代入され、それ以外の場合は 0 が代入されます。

参照

ODBC GET OPTION

ODBC GET OPTION

ODBC GET OPTION(オプション; 値)

引数	タイプ	説明
オプション	倍長整数	→ オプションの番号
値	倍長整数	← オプションの値

ODBC GET OPTION コマンドは、<オプション>に渡したオプションの現在の<値>を返します。

各種オプションとその関連する値についての詳細は、前述の**ODBC SET OPTION** コマンドを参照してください。

コマンドが正常に終了すると、システム変数OKには1が代入され、それ以外の場合は0が代入されます。

参照

ODBC SET OPTION

ODBC EXECUTE

ODBC EXECUTE(sql 命令 {;バインドオブジェクト1; ...;バインドオブジェクト_n})

引数	タイプ	説明
SQL 命令	テキスト	→ 実行する SQL コマンド
バインドオブジェクト 1..n	変数!フィールド	← 結果を受け取る (必要な場合)

ODBC EXECUTE コマンドは、SQL コマンドを実行してその結果を4Dオブジェクト（配列、変数、フィールド）にバインドするために使用します。

このコマンドを実行するには、カレントプロセス内で有効な接続が必要になります。

引数<SQL 命令>には、実行する SQL コマンドを代入し、その結果は<バインドオブジェクト>へ返されます。

各変数はカラムの並び順でバインドされます。つまり、リモートカラムのうち余ったカラムがあれば、それは切り捨てられます。

引数として4Dフィールドが<バインドオブジェクト>に渡された場合、コマンドはレコードを作成してそれを自動的に保存します。

4Dフィールドは同じテーブルに属していなければなりません。テーブル1のフィールドとテーブル2のフィールドを同じ呼び出しで渡すことはできません。別々のテーブルのフィールドを渡すと、エラーが生成されます。

引数<バインドオブジェクト>に4D配列や変数を渡す場合、処理されるデータのタイプを検証するため、コマンドを呼び出す前にこれらを宣言するようお勧めします。

4D変数を使用すると、一度に一つのレコードが取り出されます。

- ▼ この例題では、外部データソースの emp テーブルの ename カラムを取得します。結果は、4D フィールドである “[従業員]名前” に保存されます。4D レコードは自動的に作成されます。

```
SQLStmt:="SELECT ename FROM emp"
ODBC EXECUTE(SQLStmt:[従業員]名前)
ODBC LOAD RECORD(ODBC All Records)
```

- ▼ レコード作成を検証するため、トランザクション内にコードを組み込み、操作が十分であることが分かった場合にのみ確定することができます。

```
ODBC LOGIN("mysql";"root";"")
SQLStmt:="SELECT alpha_field FROM app_testTable"
START TRANSACTION
ODBC EXECUTE(SQLStmt:[Table 2]Field1)
While(Not(ODBC End Selection))
    ODBC LOAD RECORD
```

...`データ確定用のコードをここに記述する

```
End while
VALIDATE TRANSACTION` トランザクションの確定
```

- ▼ この例題では、外部データソース emp テーブルの ename カラムを取得します。その結果は、配列 aName に保存されます。レコードは 10 件ずつ取り出します。

```
ARRAY STRING(30;tNames;20)
SQLStmt:="SELECT ename FROM emp"
ODBC EXECUTE(SQLStmt;aName)
While(Not(ODBC End Selection))
    ODBC LOAD RECORD(10)
End while
```

- ▼ この例題では、外部データソースの特定の ID に対し（WHERE 句）、emp テーブルの ename カラムと job カラムを取得します。その結果は、4D 変数 vName と vJob に保存されます。最初のレコードだけを要求します。

```
SQLStmt:="SELECT ename, job FROM emp WHERE id= 3"
ODBC EXECUTE(SQLStmt; vName; vJob)
ODBC LOAD RECORD
```

- ▼ この例題では、外部データソースにおける Test テーブルの Blob_Field カラムを取得します。その結果は、BLOB 変数に保存され、その値はレコードがロードされるたびに更新されます。

```
C_BLOB(MyBlob)
ODBC LOGIN
```

```

ODBC EXECUTE("SELECT Blob_Field FROM Test";MyBlob)
While(Not(ODBC End selection))
    `結果をブラウズする
ODBC LOAD RECORD
    `呼び出しの度に MyBlob の値が更新される
End while

```

ODBC End selection

ODBC End selection → ブール

引数	タイプ	説明
		このコマンドに引数は必要ありません。
戻り値	ブール	← 結果セットの限界に達した

ODBC End selection 関数は、結果セットの限界に達したかどうかを判定するために使用します。

▼ 次のコードは、以下のパラメータを用いて外部データソース（Oracle）へ接続します。

```

C_TEXT(vName)
ODBC LOGIN("TestOracle";"scott";"tiger")
If(OK=1)
    ODBC EXECUTE("SELECT ename FROM emp";vName)
    While(Not(ODBC End selection))
        ODBC LOAD RECORD(0)
    End while
ODBC LOGOUT
End if

```

このコードは、Emp という名前のテーブルに保存された emp 名（ename）を 4D 変数である vName に返します。

ODBC LOAD RECORD

ODBC LOAD RECORD{(レコード数)}

引数	タイプ	説明
レコード数	整数	→ ロードするレコードの数

ODBC LOAD RECORD コマンドは、現在の接続において開かれた ODBC ソースからのレコードを 1 件以上 4th Dimension 内に取り込みます。

任意の引数<レコード数>を使用し、取り出すレコード数を設定します。

- この引数を省略すると、コマンドはデータソースのカレントレコードを取り出します。この方法は、一度に1レコードを受け取るループ中のデータ取得に対応します。
- <レコード数>に整数値を渡すと、コマンドは<レコード数>件のレコードを取り出します。
- 「ODBC All Records」定数（または値-1）を渡すと、コマンドはテーブルの全レコードを取り出します。

注：最後の2つの引数は、取得したデータが配列や4Dフィールドに関連付けられている場合にのみ意味を持ちます。

ODBC CANCEL LOAD

ODBC CANCEL LOAD

引数	タイプ	説明
----	-----	----

このコマンドに引数は必要ありません。

ODBC CANCEL LOAD コマンドは現在の **SELECT** リクエストを終了してパラメータを初期化します。

このコマンドを使用して、**ODBC LOGIN** コマンドにより開始された同一接続内（つまり同一カーソル）において、複数の **SELECT** リクエストを実行します

▼ この例題では、同一接続内で2つのリクエストが実行されます。

```

C_BLOB(MyBlob)
C_TEXT(MyText)
ODBC LOGIN("mysql","root","")
SQLStmt:="SELECT blob_field FROM app_testTable"
ODBC EXECUTE(SQLStmt;MyBlob)
While(Not(ODBC End selection))
    ODBC LOAD RECORD
End while
    `カーソルのリセット
ODBC CANCEL LOAD
SQLStmt:="SELECT Name FROM Employee"
ODBC EXECUTE(SQLStmt;MyText)
While(Not(ODBC End selection))
    ODBC LOAD RECORD
End while

```

ODBC SET PARAMETER

ODBC SET PARAMETER(オブジェクト; パラメータタイプ)

引数	タイプ	説明
オブジェクト	4Dオブジェクト	→ 使用される4Dオブジェクト (変数、配列、フィールド)
パラメータタイプ	倍長整数	→ パラメータのタイプ

ODBC SET PARAMETER コマンドを使用すると、4D変数や配列、フィールドをODBCリクエストで使用することができます。

ODBC リクエストで 4th Dimension オブジェクトを使用する

4th Dimension では、2通りの方法で4DオブジェクトをODBCリクエストに挿入することができます。つまり、直接関連付ける方法と、**ODBC SET PARAMETER** を使用してパラメータを指定する方法です。

■ 直接関連付ける方法：このモードでは、リクエストテキスト内の“<<”と“>>”記号の間に、使用する4Dオブジェクト (変数、配列、フィールド) の名前を挿入する必要があります。

例：INSERT INTO emp(empno,ename) VALUES(<<vEmpno>>,<<vEname>>)

この例では、リクエストの実行中に4D変数であるvEmpnoとvEnameの値でパラメータが置き換えられます。このソリューションは、4Dフィールドと配列を使用した場合でも有効です。

この使いやすいシンタックスには、SQL規格に準拠していないという難点が1つあり、またexitパラメータを使用することもできません。この問題を改善するために、**ODBC SET PARAMETER** コマンド (二番目の方法) を使用することができます。

■ パラメータを指定する方法：この方法では、**ODBC SET PARAMETER** コマンドを使用して、ODBCリクエスト内で使用したい4Dオブジェクトを指定することができます。

■ 引数<オブジェクト>には、リクエストで使用される4Dオブジェクト (変数、配列、フィールド) を渡します。

■ 引数<パラメータタイプ>には、パラメータのSQLタイプを渡します。任意の値を渡すか、「External Data Source」テーマにある次の定数のいずれかを使用することができます。

定数	タイプ	値
OBDC Param In	倍長整数	1
OBDC Param In Out	倍長整数	2
OBDC Param Out	倍長整数	4

SQL リクエスト内におかれた “?” 記号が4D オブジェクトの値で置き換えられます (標準のシンタックス)。

リクエストに複数の “?” 記号が含まれている場合、**ODBC SET PARAMETER** コマンドを複数回呼び出す必要があります。4D オブジェクトの値は、コマンドの実行順に合わせてリクエスト内で順次割り当てられます。

- ▼ この例題を使用して、関連付けられた4D変数を直接呼び出すODBCリクエストを実行します。

```

C_TEXT(MyText)
C_LONGINT(MyLongint)
ODBC LOGIN("mysql";"root";"")
SQLStmt:="insert into app_testTable(alpha_field, longint_field) VALUES
                                                (<<MyText>>, <<MyLongint>>)"
For(vCounter;1;10)
    MyText:="Text"+String(vCounter)
    MyLongint:=vCounter
    ODBC EXECUTE(SQLStmt)
End for

```

- ▼ 上記と同じ例題ですが、**ODBC SET PARAMETER** コマンドが使用されています。

```

C_TEXT(MyText)
C_LONGINT(MyLongint)
ODBC LOGIN("mysql";"root";"")
SQLStmt:="insert into app_testTable(alpha_field, longint_field) VALUES(?,?)"
For(vCounter;1;10)
    MyText:="Text"+String(vCounter)
    MyLongint:=vCounter
    ODBC SET PARAMETER(MyText;ODBC Param In)
    ODBC SET PARAMETER(MyLongint;ODBC Param In)
    ODBC EXECUTE(SQLStmt)
End for

```

- ▼ この例題を使用して、関連付けられた4D配列を直接使用するODBCリクエストを実行します。

```

ARRAY TEXT(MyTextArray;10)
ARRAY LONGINT(MyLongintArray;10)

```

```

For(vCounter;1;Size of array(MyTextArray))
    MyTextArray{vCounter}:="Text"+String(vCounter)
    MyLongintArray{vCounter}:=vCounter
End for
ODBC LOGIN("mysql","root","")
SQLStmt:="insert into app_testTable(alpha_field, longint_field) VALUES
    (<<MyTextArray>>, <<MyLongintArray>>)"
ODBC EXECUTE(SQLStmt)

```

▼ この例題を使用して、関連付けられた4Dフィールドを直接使用する ODBC リクエストを実行します。

```

ALL RECORDS([Table 2])
ODBC LOGIN("mysql","root","")
SQLStmt:="insert into app_testTable(alpha_field, longint_field) VALUES
    (<<[Table 2]Field1>"+", <<[Table 2]Field2>)"
ODBC EXECUTE(SQLStmt)

```

ODBC GET LAST ERROR

ODBC GET LAST ERROR(エラーコード;エラーテキスト;ODBCエラー;SQL Serverエラー)

引数	タイプ	説明
エラーコード	倍長整数	← エラーコード
エラーテキスト	倍長整数	← エラーテキスト
ODBCエラー	テキスト	← ODBCエラーコード
SQLServerエラー	テキスト	← SQLサーバのネイティブエラーコード

ODBC GET LAST ERROR コマンドは、ODBCコマンドの実行中に発生した最後のエラーに関連する情報を返します。エラーの発生箇所としては、4th Dimension アプリケーション、ネットワーク、ODBCソースなどが考えられます。

通常、このコマンドは **ON ERR CALL** コマンドを用いて設定されたエラー処理用メソッド内から呼び出されます。

- 引数<エラーコード>はエラーコードを返します。
- 引数<エラーテキスト>はエラーテキストを返します。

最後の2つの引数は、ODBCソースからのエラーの場合にのみ代入されます。それ以外の場合は、空のテキストが返されます。

- 引数<ODBCエラー>はODBCエラーコード (SQL state) を返します。
- 引数<SQLServerエラー>はSQLサーバのネイティブエラーコードを返します。

ODBC IMPORT

ODBC IMPORT(ソーステーブル{; プロジェクト}{; *})

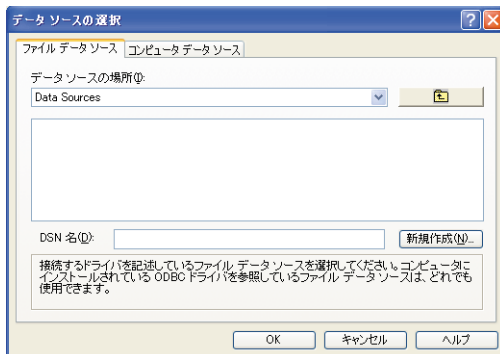
引数	タイプ	説明
ソーステーブル	文字列	→ ODBC データソース内のテーブル名
プロジェクト	BLOB →	読み込むプロジェクトの内容
		← 読み込むプロジェクトの新しい内容 (“*” を渡した場合)
*	*	→ 読み込み用ダイアログボックスの表示とプロジェクトの更新

ODBC IMPORT コマンドを使用すると、外部 ODBC ソースのテーブル <ソーステーブル> からデータが読み込まれます。接続パラメータ（ソース名、ユーザとパスワード）は、BLOB <プロジェクト> に納められます。

注：プロジェクトには、すべての読み込みパラメータ（特に、データソース、対象のテーブルやフィールド）が格納されます。ODBC 読み込み用ダイアログボックスでこれらのパラメータを設定した後、必要に応じてこれらのパラメータをディスク上のフィールドに保存することができます。詳細については、『ユーザリファレンス』マニュアルを参照してください。

任意の引数 <プロジェクト> を渡さない場合、**ODBC IMPORT** コマンドは ODBC データソースを選択するダイアログボックスを表示します。

Windows



Mac OS



ソースを選択すると、4th Dimension の ODBC 読み込み用ダイアログボックスが現れ、操作の設定を行うことができます。2つのダイアログボックスのいずれかで「キャンセル」ボタンをクリックすると、実行は中断されてシステム変数 OK に 0 が代入されます。

引数<プロジェクト>に有効な ODBC 読み込みプロジェクトを格納した BLOB を渡すと、ユーザによる操作を必要とせずに直接読み込みが実行されます。これを行うには、**DOCUMENT TO BLOB** コマンドを使用し、あらかじめディスク上に保存されたプロジェクトを引数<プロジェクト>に渡したフィールドや BLOB 変数内にロードしておく必要があります。

また、空の引数<プロジェクト>と任意の引数<*>を用いて **ODBC IMPORT** コマンドを呼び出した後で、引数<プロジェクト>を BLOB フィールド（後述）に保存することもできます。

一方、このソリューションにより、データファイルを用いてプロジェクトを保存し、他方ではディスクから BLOB へのプロジェクトの保存フェーズを回避することができます。

任意の引数<*>を指定した場合、<プロジェクト>に指定された設定（存在する場合）を用いて、ODBC 読み込み用のダイアログボックスが表示されます。これにより、定義済みのプロジェクトを使用しながらも、いくつかのパラメータを変更することができます。さらに、その際はダイアログボックスのクローズ後に、“新規”プロジェクトのパラメータが引数<プロジェクト>へ格納されます。この後、それを BLOB フィールドやディスクファイルなどに保存することができます。

読み込みが正常に実行されると、システム変数 OK に 1 が代入されます。

参照

ODBC EXPORT

ODBC EXPORT

ODBC EXPORT(ソーステーブル{; プロジェクト}{; *})

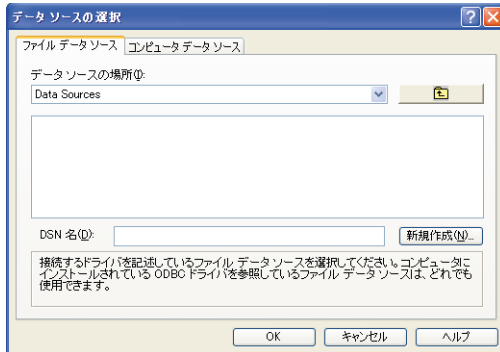
引数	タイプ	説明
ソーステーブル	文字列	→ ODBC データソース内のテーブル名
プロジェクト	BLOB →	書き出すプロジェクトの内容
		← 書き出すプロジェクトの新しい内容 (“*” を渡した場合)
*	*	→ 書き出し用ダイアログボックスの表示とプロジェクトの更新

ODBC EXPORT コマンドを使用すると、外部 ODBC ソースのテーブル<ソーステーブル>へデータが書き出されます。接続パラメータ（ソース名、ユーザとパスワード）は、BLOB <プロジェクト>に納められます。

注：プロジェクトには、すべての書き出しパラメータ（特に、データソース、書き出されるテーブルやフィールド）が格納されます。ODBC書き出し用ダイアログボックスでこれらのパラメータを設定した後、必要に応じてこれらのパラメータをディスクファイルに保存することができます。詳細については、『ユーザリファレンス』マニュアルを参照してください。

任意の引数<プロジェクト>を渡さない場合、**ODBC EXPORT** コマンドはODBCデータソースを選択するダイアログボックスを表示します。

Windows



Mac OS



ソースを選択すると、4th DimensionのODBC書き出し用ダイアログボックスが現れ、操作の設定を行うことができます。2つのダイアログボックスのいずれかで「キャンセル」ボタンをクリックすると、実行は中断されてシステム変数OKに0が代入されます。

引数<プロジェクト>に有効なODBC書き出しプロジェクトを格納したBLOBを渡すと、ユーザによる操作を必要とせずに直接書き出しが実行されます。これを行うには、**DOCUMENT TO BLOB** コマンドを使用し、あらかじめディスク上に保存されたプロジェクトを引数<プロジェクト>に渡したフィールドやBLOB変数内にロードしておく必要があります。

また、空の引数<プロジェクト>と任意の引数<*>を用いて**ODBC EXPORT** コマンドを呼び出した後で、引数<プロジェクト>をBLOBフィールド（後述）に保存することもできます。

任意の引数<*>を指定した場合、<プロジェクト>に指定された設定（存在する場合）を用いて、ODBC書き出し用のダイアログボックスが表示されます。これにより、定義済みのプロジェクトを使用しながらも、いくつかのパラメータを変更することができます。さらに、その際はダイアログボックスのクローズ後に、“新規”プロジェクトのパラメータが引数<プロジェクト>へ格納されます。この後、それをBLOBフィールドやディスクファイルなどに保存することができます。

読み込みが正常に実行されると、システム変数OKに1が代入されます。

参照

ODBC IMPORT

ツール

新しい「ツール」テーマには各種ユーティリティコマンドが再編成されています。

- **BUILD APPLICATION** : このコマンドを使用すると、アプリケーションを生成する自動化されたメソッドを導入することができます。
- **ENCODE** および **DECODE** : Base64 フォーマットでの BLOB のエンコードとデコードを可能にします。
- **SPELL CHECKING** および **SET DICTIONARY** : スペルチェック機能を使用します。

BUILD APPLICATION

BUILD APPLICATION{(プロジェクト名)}

引数	タイプ	説明
プロジェクト名	文字列	→ 使用するプロジェクトのフルアクセスパス

BUILD APPLICATION コマンドは、アプリケーション生成プロセスを開始します。現行のアプリケーションプロジェクト、または引数<プロジェクト名>で指定されたアプリケーションプロジェクト内に設定されたパラメータが考慮されます。

アプリケーションプロジェクトは XML ファイルであり、アプリケーション生成に使用されるすべてのパラメータが格納されます。パラメータの大半は、「アプリケーションビルド...」ダイアログボックス上に表示されます（詳細は、前述の「アプリケーションビルド」の節を参照）。デフォルトとして、4th Dimension は各データベースに対し “buildapp.xml” という名前のアプリケーションプロジェクトを作成し、それをデータベースの「Preferences」フォルダ内の「BuildApp」サブフォルダに配置します。

データベースがコンパイルされていない場合や、コンパイルされたコードが古い場合、コマンドはまず最初にコンパイルプロセスを開始します。その際、コンパイラウィンドウは表示されず（エラー発生時を除く）、進捗バーだけが表示されます。

任意の引数<プロジェクト名>を渡さない場合、コマンドは現行のアプリケーションプロジェクトに指定されたパラメータを使用します。ダイアログボックスは表示されません。

有効なアプリケーションプロジェクト用の XML ファイル（拡張子 “.xml”）のアクセスパスと名前を渡した場合、コマンドはこのファイルに指定されたパラメータを使用します。

アプリケーションプロジェクトのXMLファイルにおいて使用可能な構造とキーに関する詳細は、4th Dimensionの『XML キー』マニュアル（HTML形式で提供）を参照してください。

コマンドが正常に実行されると、システム変数OKに1が代入され、それ以外の場合は0が代入されます。後者の場合はエラーが生成され、このエラーは**ON ERR CALL**コマンドを用いて遮ることができます。

▼ この例題は、1つのメソッド内で2つのアプリケーションを構築します。

```
BUILD APPLICATION("c:¥¥folder¥¥projects¥¥myproject1.xml")
If(OK=1)
    BUILD APPLICATION("c:¥¥folder¥¥projects¥¥myproject2.xml")
End if
```

ENCODE

ENCODE(blob)

引数	タイプ	説明
blob	BLOB	→ Base64 フォーマットにエンコードする BLOB
		← Base64 フォーマットにエンコードされた BLOB

ENCODE コマンドは、BLOB を Base64 フォーマットにエンコードします。コマンドは、引数として渡された < blob > を直接修正します。

Base64 エンコーディングは、7桁を超える有効ビットを保持しないように、8ビットにコード化されたデータを修正します。例えば、XML を用いた BLOB の処理にはこのエンコードが必要になります。

参照

DECODE

DECODE

DECODE(blob)

引数	タイプ	説明
blob	BLOB	→ Base64 フォーマットにエンコードされた BLOB
		← デコードされた BLOB

DECODE コマンドを使用すると、引数< blob >に渡されたBase64 フォーマットにエンコード済みのBLOBをデコードすることができます。コマンドは、引数として渡された< blob >を直接修正します。

コマンドは< blob >の内容を検証しません。渡したデータが実際にBase64 フォーマットでエンコードされているかどうかを確認しなくてはなりません。これを行わない場合は結果が正しくなくなります。

参照

ENCODE

SPELL CHECKING

SPELL CHECKING

引数	タイプ	説明
----	-----	----

このコマンドに引数は必要ありません。

SPELL CHECKING コマンドをは、現在表示中のフォーム上でフォーカスを取得しているフィールドや変数のスペルチェックを開始します。

チェックされるオブジェクトは、文字列またはテキストタイプでなくてはなりません。

注：フォームのボタンを用いてスペルチェックを起動したい場合、ボタンに「フォーカス表示」プロパティが設定されていないことを確認しなくてはなりません。

スペルチェックは、フィールドや変数の最初の語句から開始します。未知の語句が見つかり、スペルチェックダイアログボックスが表示されます（詳細については、前述の「スペルチェック」の節を参照）。**SET DICTIONARY** コマンドを使用しない限り、4th Dimension はカレント辞書（アプリケーションの言語に対応）を使用します。

参照

SET DICTIONARY

SET DICTIONARY

SET DICTIONARY(辞書)

引数	タイプ	説明
辞書	倍長整数	→ スペルチェックに使用する辞書

SET DICTIONARY コマンドは、カレント辞書を引数< 辞書 >で指定された辞書で置き換えます。

カレント辞書は、4th Dimensionの統合スペルチェック機能で使用されます（詳細は、前述の「スペルチェック」の節を参照）。デフォルトとして、4th Dimensionはアプリケーション言語に対応する辞書を使用します。フランス語、英語、ドイツ語、スペイン語の4つの辞書が使用できます。

<辞書>には、使用する辞書の番号を渡します。「Dictionaries」テーマ内にある次の定義済み定数のいずれかを使用することができます。

定数	タイプ	値
English Dictionary	倍長整数	69632
German Dictionary	倍長整数	131584
Spanish Dictionary	倍長整数	196608
French Dictionary	倍長整数	262144

辞書が正常にロードされると、システム変数OKに1が代入され、それ以外の場合は0が代入されてエラーが返されます。

参照

SPELL CHECKING

記号

##D	46
&	36
.4BK	197
.4BL	198
.4DA	243
.xml	406

数字

3D ボタン

アイコンを追加	153
カスタム	155
タイトル／ピクチャ位置	154
フォーマットを適用する	308
変更	148
「ポップアップメニュー」プロパティ	154

3D ボタンにアイコンを追加する 153

3 ステートチェックボックス 148

4D Backup 226

互換性 226

サポートされない4D Backupの機能 226

メリット 227

4D Customizer Plus 62

4D Engine 174

4D Insiderのグループ 101

4D Runtime Classic 174

4D Runtime Interpreted 174

4D Runtime Single User 174

4D Runtime Volume License 174

4D Syntax.rsr 27

4D コードマクロの呼び出し 102

A

APPEND TO ARRAY 354

APPLY XSLT TRANSFORMATION 387

AUTOMATIC RELATIONS 280

B

BACKUP 356

BLOB TO USERS 260

BUILD APPLICATION 406

buildapp.XML 175

C

CFM フォーマット 17

CGIのサポート 352

CHANGE ACCESS 262

CHANGE CURRENT USER 262

Classic モード 17

Count in array 354

Count list items 256

CREATE USER FORM 245

Current form page 274

Current form window 266

D

DECODE 407

DELETE LISTBOX COLUMN 319

DELETE LISTBOX ROW 325

DELETE USER FORM 246

DIALOG 301

DISPLAY NOTIFICATION 300

DISPLAY SELECTION 345

Displayed line number 343

DOC モードでの公開 185

DOC モードのサポート拡張 186

DOM Create XML element 379

DOM Create XML Ref 377

DOM EXPORT TO FILE 383

DOM EXPORT TO VAR 384

DOM Find XML element 384

DOM GET XML ELEMENT VALUE 385

DOM REMOVE XML ELEMENT 382

DOM SET XML ATTRIBUTE 378

DOM SET XML ELEMENT NAME	381	H	
DOM SET XML ELEMENT VALUE	382	HIGHLIGHT RECORDS	344
DOM SET XML OPTIONS	376	HTTPSのポート番号を変更	231
E			
EDIT FORM	244	I	
EDIT FORMULA	241	INPUT FORM	279
EDIT ITEM	296	INSERT LISTBOX COLUMN	318
ENCODE	407	INSERT LISTBOX ROW	324
Extra	173	L	
F			
FIRST PAGE	274	Last object	295
Focus object	295, 316	LAST PAGE	274
G			
GET ALLOWED METHODS	240	LAUNCH EXTERNAL PROCESS	339
GET AUTOMATIC RELATIONS	286	List item parent	255
GET BACKUP INFORMATION	357	LIST USER FORMS	246
Get default user	261	M	
GET FIELD RELATION	281	Mac OS 9	17
GET FIELD TITLES	293	Mac OS X上でWebサーバを開始する	233
GET FORM OBJECTS	278	Mach-O	17
GET FORM PROPERTIES	274	MODIFY SELECTION	345
Get format	306	MOVED LISTBOX COLUMN NUMBER	323
GET HIGHLIGHTED RECORDS	342	MOVED LISTBOX ROW NUMBER	327
GET HTTP BODY	348	N	
GET LIST ITEM PROPERTIES	253	NEXT PAGE	274
GET LIST PROPERTIES	251	O	
GET LISTBOX ARRAYS	328	ODBC CANCEL LOAD	399
Get listbox column width	322	ODBC End selection	398
Get listbox information	329	ODBC EXECUTE	396
Get listbox rows height	327	ODBC EXPORT	404
Get number of listbox columns	320	ODBC GET LAST ERROR	402
Get number of listbox rows	326	ODBC GET OPTION	395
Get plugin access	259	ODBC IMPORT	403
GET PRINT OPTION	299	ODBC LOAD RECORD	398
GET RESTORE INFORMATION	358	ODBC LOGIN	393
GET SERIAL PORT MAPPING	355	ODBC LOGOUT	394
GET TABLE TITLES	293	ODBC SET OPTION	395
GET USER PROPERTIES	264	ODBC SET PARAMETER	400
GET WEB FORM VARIABLES	352	ODBC コマンド	392
GET XSLT ERROR	390	ODBC データソース	
GOTO PAGE	274	定義	38
GOTO SELECTED RECORD	346	データ書き出し	41
		データ読み込み	39
		ODBC リクエスト	400
		On After Sort (フォームイベント)	290

- On arrow click (フォームイベント)288
 「On backup shutdown」データベースメソッド
359
 「On Backup Startup」データベースメソッド ..
359
 On Before Data Entry (フォームイベント)・289
 On Collapse (フォームイベント).....292
 On Column Moved (フォームイベント) ...290
 On Column Resize (フォームイベント) ...290
 On expand (フォームイベント)292
 On Header Click (フォームイベント)290
 On Load Record (フォームイベント)291
 On long click (フォームイベント)288
 On Mouse Enter (フォームイベント)287
 On Mouse Leave (フォームイベント)287
 On Mouse Move (フォームイベント)287
 On Row Moved (フォームイベント)290
 On Selection Change (フォームイベント)
289, 291
 OPEN 4D PREFERENCES332
 OS15
 OUTPUT FORM280
- P**
- Pop up menu295
 Pop up window271
 PREVIOUS PAGE274
 PRINT SELECTION中にレコード毎にフォーム
 をリロード56
 PROCESS HTML TAGS347
- Q**
- qr cmd standard deviation45
 QR EXECUTE COMMAND45
 QR Get command status45
 QR GET TOTALS DATA45
 QR SET TOTALS DATA45, 46
- R**
- RAM16
 Resizable sheet window273
 RESIZE FORM WINDOW267
 RESTORE356
- S**
- SAX ADD PROCESSING INSTRUCTION ..368
 SAX ADD XML CDATA367
 SAX ADD XML COMMENT363
 SAX ADD XML DOCTYPE363
 SAX ADD XML ELEMENT VALUE366
 SAX CLOSE XML ELEMENT366
 SAX GET XML CDATA373
 SAX GET XML COMMENT370
 SAX GET XML DOCUMENT VALUES370
 SAX GET XML ELEMENT371
 SAX GET XML ELEMENT VALUE372
 SAX GET XML ENTITY374
 SAX Get XML node369
 SAX GET XML PROCESSING INSTRUCTION ..
371
 SAX OPEN XML ELEMENT364
 SAX OPEN XML ELEMENT ARRAYS365
 SAX SET XML OPTIONS362
 SCROLL LINES294
 Select document302
 SELECT LIST ITEM248
 SELECT LIST ITEMS BY POSITION248
 SELECT LIST ITEMS BY REFERENCE
249, 256
 SELECT LISTBOX ROW323
 Selected list item255
 Selected list items255
 SEND HTTP RAW DATA349
 SET ALLOWED METHODS239
 SET AUTOMATIC RELATIONS280
 SET CGI EXECUTABLE352
 SET COLOR312
 SET DICTIONARY408
 SET ENTERABLE307
 SET ENVIRONMENT VARIABLE339
 SET FIELD RELATION284
 SET FORM HORIZONTAL RESIZING276
 SET FORM SIZE274
 SET FORM VERTICAL RESIZING277
 SET FORMAT308
 SET LIST ITEM254
 SET LIST ITEM PROPERTIES252
 SET LIST PROPERTIES250
 SET LISTBOX COLUMN WIDTH322
 SET LISTBOX GRID COLOR331
 SET LISTBOX ROWS HEIGHT326
 SET PLUGIN ACCESS258
 SET PRINT OPTION299

SET RGB COLORS	310
Set user properties	264
SET WEB SERVICE OPTION	390
SET WINDOW RECT	268
SET XSLT PARAMETER	389
Sheet window	273
SHOW LISTBOX GRID	330
SHOW LISTBOX SCROLLBAR	332
SOAP ライセンス	185
SORT LISTBOX COLUMNS	321
SPELL CHECKING	408
T	
TCPポート番号	65
TRIC リソース	64
U	
URL	232
USERS TO BLOB	259
V	
Validate password	263
W	
Web サーバの検証	234
Web サーバ	
Mac OS X上で開始する	233
検証	234
未知の URL	56, 232
Web サービス	184
DOC モードでの公開	185
DOC モードのサポート拡張	186
Web サービスの利用	185
リクエストにオプションを設定する	187
WEDD	63
X	
XML	
DOM コマンドを用いてXMLを設定する	374
SAX を介したXML 文書の作成	361
エラーコード	385
パラメータファイル	225
XML キー	184
XPath ノーテーション	375
XSLT	386
XSL 言語	386

Z	
! (OR)	36
あ	
空きメモリ中キャッシュに使用する率	59
アクセス管理	265
アダプティブ (適応) キャッシュの計算	58
新しいラジオボタンの動作	157
圧縮率	204
アップデート通知の表示	183
アプリケーションビルダ	173
「アプリケーションビルド」ダイアログ	
ボックス	174
ログファイル	176
い	
移動	112
印刷設定	115
インスペクタ	73
インタフェース	
ツールバー	67
ツールボックス	74
メニューの再編成	69
インターレース率	204
え	
エクスプローラ	86
オブジェクトの回復	88
オブジェクトの削除	96
オブジェクトの作成と変更	96
オブジェクトを削除	88
ゴミ箱ページ	87
ゴミ箱を空にする	89
コンポーネントページ	90
フォルダを使用する	92
演算子	
お	
オブジェクト	
回復	88
削除	88
表示	31
編集	31
オブジェクトの作成時にフォルダへ追加する	98
オブジェクトの削除	88, 96
オブジェクトをビューに配置する	118

オブジェクトライブラリ	168	アップデート通知の表示	183
オブジェクトライブラリの使用	34	強制的に自動更新を行う	184
クライアント/サーバの対応	173	クライアント/サーバ	177
構築	170	「クライアント/サーバ」タブ	175
コンテキストメニュー	171	クライアント/サーバアプリケーションを構築する	176
作成	168	アプリケーションパラメータを設定	177
従属オブジェクト	172	生成されるファイル	179
表示	172	「クリップボードへコピー」ボタン	45
開く	169	グループ化/グループ解除	113
か		け	
解除	26	警告	86
現在のデータベース	26	検索	68
すべての4D Server サービス	26	こ	
回復		公開	66
オブジェクト	88	公開情報	66
概要	13	「コード作成」ボタン	45
書き出し		コードの生成	43
ODBC ソースへの書き出し	41	互換性	15
ODBC データソースを使用	37	コマンドラインインタフェース	23
XSL ファイルへのリンク挿入	37	ゴミ箱を空にする	89
XSL 変換を適用する	37	コンパイラ	86
カスタム3D ボタン	155	さ	
カスタムテンプレートのクリア	57	サーバアプリケーションの更新	183
画面解像度	16	サービス	
空の時間	36	4D Server を登録する	25
空の日付	36	最小サイズ	60
空の文字列	36	最大サイズ	60
環境設定	52	サイドプッシャー	269
き		削除 (バックアップファイルの設定)	196
キーボードによる移動単位	57	作成時の設定へ戻す	30
境界線	44	サブフォームオプション	165
強制的に更新	184	サイズ調整	167
許可-拒否テーブル設定	66	水平スクロールバー	165
均等配置ボタン	113	ダブルクリックアクション	165
く		フォーカス	167
クイックレポートエディタ		サポートされない4D Backup の機能	226
「4D コード作成」ボタン	43	し	
新しい演算子	45	シートウインドウ	272
境界線	44	次回に予定された日付と時刻に再試行する	
新機能	43	(バックアップの設定)	203
標準偏差	45	時間ごと (定期バックアップ)	205
組み込み参照の計算	156	指数	36
クライアントアプリケーションの自動更新	183		

持続的接続	229	設定時間待った後、バックアップを中止する	202
指定時間経過後に再試行	203	選択ツール	112
自動クライアント再接続機能	61	た	
自動スクロール	57	ダイアログボックスでフィールドを入力不可にする	55
自動整列	58	タイムアウト	231
自動バックアップ	206	つ	
自動復元	217	追加 (バックアップファイルの設定)	196
最新バックアップから復元する	218	ツールバー	67
自動復元の後、データベースを開始する	219	ツールボックス	74
ログを統合する	219	グループ	77
週ごと (定期バックアップ)	206	スタイルシート	83
手動でバックアップを復元する	220	ピクチャ	79
手動でログを統合する	222	フィルタ	83
使用ログファイル (バックアップの設定)	210, 213	メニュー	78
ショートカット	63, 72	ユーザ	76
処理の取り消し	223	リスト	81
シリアル番号の管理	182	月ごと (定期バックアップ)	206
シンタックスヘルプファイル	26	次のバックアップ予定 (バックアップファイルの設定)	199
す		常に処理の終了を待つ	202
ズーム	112, 116	常に表示	163
スクリプトマネージャ	64	て	
「スタンドアロン」タブ	175	定期的バックアップの設定	205
ステータス (バックアップファイルの設定)	198	データの読み込みと書き出し	36
ストラクチャファイル (バックアップファイルの設定)	196	データファイル (バックアップファイルの設定)	195
スプラッシュスクリーンを表示する	62	データファイルが修正された時にバックアップ	201
スプリッタ	140	データベースが完全でない場合、最新のログを統合する	219
ウインドウ	269	データベース名	197
反応領域	141	テーブル番号	73
プッシュアッププロパティ	141	テキストプロパティ	44
ページ0のスプリッタ	141	「デフォルトで非表示」プロパティ	157
リアルタイム表示と「On click」イベント	142	デフォルト表示	58
せ		デフォルトフォルダ	94
整数による除算	36	テンプレートとして使用	123
整列	113	と	
セグメント		統合されたオブジェクトバーとツールバー	111
サイズ	204	オブジェクトバー	114
番号付け範囲	198	ツールバー	111
接続毎のリクエスト数	231		

- 動作環境15
- 同封ファイル (バックアップファイルの設定)196
- 同封ファイルが見つからない194
- 登録25
 - 4D Server25
- ドラッグ&ドロップ開始57
- トランザクション208
- に
- 入力順序112
- ね
- ネイティブな入力制御144
- は
- バージョン6.7.x15
- バージョン6.8.x15
- はじめに13
- パスワード (変更)48
- バックアップ
 - 圧縮率204
 - インターレース率204
 - 開始191
 - 環境設定190
 - 管理190
 - 最後のバックアップ情報198
 - 実行192
 - 自動バックアップ206
 - 自動復元217
 - ジャーナル224
 - 手動191
 - 手動で復元する220
 - 手動でログを統合する222
 - 障害216
 - 処理の取り消し223
 - セグメント化204
 - セグメントサイズ204
 - 定期的な自動バックアップ191
 - 定期的バックアップの設定205
 - データの復元216
 - バックアップ中にデータベースへアクセスする193
 - バックアップに使用できる容量201
 - パラメータ195
 - 番号付け範囲198
 - ファイル名197
 - 保存する場所197
 - 問題が発生した場合194
 - リダンダンシー率204
 - バックアップ不可能194
 - バックアップジャーナル224
 - バックアップ設定
 - アーカイブ203
 - 最新のバックアップのみ保存する200
 - 設定時間待った後、バックアップを中止する202
 - データファイルが更新された場合のみバックアップを行う201
 - トランザクション中またはインデックス処理中202
 - バックアップ失敗時203
 - バックアップ中にデータベースへアクセスする193
 - バックアップパラメータ195
 - 最後のバックアップ情報198
 - バックアップの内容195
 - バックアップファイル名197
 - 保存先197
 - バックアップファイル名 (バックアップファイルの設定)199
 - バックアップモジュール189
 - バッジ121
 - 表示114
 - パレット表示44
- ひ
- 日ごと (定期バックアップ)206
- 非自動バックアップ205
- 日付と時間 (バックアップファイルの設定)198
- ビュー116
 - オブジェクトを配置118
 - 管理116
 - 名前の変更119
 - ビューを使用して作業する119
- ビュー管理114, 116
- ビュー名の変更119
- 表示とページ管理113
- 標準アクション167
- 「標準偏差」演算子45

ふ	
フィールド	73
フィールド番号	73
フルタイプ	156
フォーミュラエディタ	35
インタフェース	35
演算子リスト	35
コマンドリスト	36
フォームイベント	286
3D ボタン	288
リストボックス	289
レコードリスト	291
ロールオーバー	287
フォームエディタ	107
テンプレートとして使用	123
プロパティリスト	107
「ユーザ編集可」プロパティ	126
フォームオブジェクト	128
フォームの印刷設定を編集	115
フォルダ情報	100
フォルダとサブフォルダを整理する	97
移動	98
オブジェクトの作成時にフォルダへ追加する	98
ドラッグ&ドロップ	98
フォルダの追加 (バックアップファイルの設定)	196
フォルダやオブジェクトの名前を変更する	95
フォルダやサブフォルダを作成する	94
フォルダを使用	
エクスプローラ	92
作成	94
従来の動作との互換性	94
情報	100
整理	97
デフォルトフォルダ	94
名前を変更	95
復元	216
自動復元	217
障害と診断	216
プラグイン	
「Mac4DX」または「Win4DX」フォルダ	19
管理	181
ダウンロードのメカニズム	22
場所	18
「プラグイン (PlugIns)」フォルダ	19
プラグインアーキテクチャ	18
スタンドアロン	21
内部アーキテクチャ	21
「プラグイン」タブ	175
プラグインの互換性	17
プラットフォームインタフェース	142
印刷	143
互換性とプロパティへのアクセス	143
システム	142
プリントの進捗状況を表示する	62
プロパティリスト	107
選択リストの設定	108
南京錠アイコン	110
プロパティ表示ページ	109
へ	
ペインの表示	216
変換	15
「変更」ボタン	
隠す	49
使用	48
ほ	
保存	
データベースアクセスパラメータ	50
ま	
マグネティックグリッド	125
マクロフォルダ	105
マニュアルについて	14
み	
未知のURLの“/”を取り除く	56
め	
メインメモリ	58
メソッドエディタ	101
4D コードマクロの呼び出し	102
ツールバー	106
呼び出されたメソッドの実行コンテキスト	102
メソッドのプレビューとプロパティの表示	91
メニューの再編成	69
も	
最も古いバックアップファイルの削除	202

ゆ

- ユーザストラクチャファイル (バックアップ
ファイルの設定)196
- ユーザによる中断194
- ユーザのパスワードを変更する48
- ユーザフォーム242
 - オブジェクトを挿入243
 - 管理243
 - 保存243
- ユーザフォームエディタ29
 - オブジェクトバー31
 - コンテキストメニュー33
 - ツールバー30
 - 表示と編集31
 - 「編集」メニューコマンド33
- ユーザ編集可29, 126, 243
- ユーザロック112

よ

- 呼び出されたメソッドの実行コンテキスト102
- 読み込み
 - ODBC データソースを使用37
 - ODBC ソースからの読み込み39

ら

- 「ライセンス」タブ175
- ラジオボタンを名前グループ化55
- ランタイムエクスプローラ73

り

- リストの表示161
 - アクティブオブジェクト165
 - オブジェクトの表示163
 - 空の行162
 - フォームイベントの呼び出し順序164
 - フォームのボディにあるオブジェクト162
- リストボックス
 - オブジェクトメソッドを付加する315
 - カラム固有のプロパティ135
 - 管理289
 - 固有のプロパティ132
 - ヘッダ固有のプロパティ139
 - 「ユーザ」モード130
- リダンダンシー率205

れ

- レイヤボタン113
- レコード選択時163
- レコード非選択時163
- レコードリスト158
 - 選択モード159
 - 表示161
 - リスト更新161

ろ

- ログインダイアログボックス48
 - データベースアクセスパラメータの保存50
 - パスワードの変更48
- ログバックアップファイル名 (バックアップ
ファイルの設定)199
- ログファイル176
 - ウインドウのカスタマイズ214
 - 解析213
 - 管理207
 - 機能図208
 - 作成210
 - 手動でログを統合する222
 - 処理の取り消し223
 - 操作208
 - 中止213
 - 番号付け範囲198
 - フィールドを割り当てる215
 - ペインの表示216
 - ログ名198
- ログファイルをチェック213

わ

- 割り当てる物理メモリ59

