

4th Dimension 6.5

アップグレード
Windows[®] and Mac[™] OS



ACI

4th Dimension 6.5 アップグレード Windows® and Mac™ OS

Copyright© 1999 - 2000 ACI 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 ACI 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、ACI ロゴ、およびその他の ACI 製品の名称は、ACI 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.....9 |
| | 本マニュアルについて.....9 |
| | 4D 3.x.x/6.0.x データベースとの互換性10 |
| 第2章 | デザインモード11 |
| | データベースの新規作成とオープン.....12 |
| | 新規データベースの作成.....12 |
| | 既存のデータベースを開く.....13 |
| | ストラクチャウインドウのインタフェースと |
| | ナビゲーション.....14 |
| | コンテキストメニュー.....14 |
| | ストラクチャエディタ内での選択に |
| | キーボードを使用する.....15 |
| | テーブルの並びの管理.....15 |
| | テーブルの作成.....15 |
| | 文字列フィールドの長さの表示.....16 |
| | フォーム.....16 |
| | デフォルトフォームの作成.....16 |
| | フォームエディタの変更.....17 |
| | コンテキストメニュー.....18 |
| | プロパティリスト.....29 |
| | フォームオブジェクト.....33 |
| | 出力フォームのヘッダ.....40 |
| | オンラインヘルプ.....41 |
| | ランタイムエクスプローラ.....44 |
| | ウインドウの表示.....44 |
| | ウォッチページ.....45 |
| | プロセスページ.....46 |
| | ブレークおよびキャッチページ.....47 |
| | オブジェクト検索エディタ.....47 |
| | 検索のオプション.....50 |

| | | |
|-----|----------------------------------|----|
| | コメント | 51 |
| | オブジェクトにコメントを付ける | 51 |
| | コメント用フォント属性の変更 | 52 |
| | 日付、時間またはユーザ名の挿入 | 53 |
| | 自動コメントの挿入 | 54 |
| | メソッドの編集 | 56 |
| | シンタックスヘルプ | 56 |
| | テキスト編集を操作するショートカット | 59 |
| | メソッドエディタ内でのフォントの定義 | 59 |
| | ブレークポイントの挿入 | 60 |
| | 新しいデータベースプロパティ | 61 |
| | 文字「@」の管理 | 61 |
| | テンポラリフォルダをどこに置くかを定義する | 62 |
| | スタートアップ時にクライアントを登録する | 63 |
| | パスワード | 64 |
| | ユーザモードへのグループのアクセス | 64 |
| | デフォルトユーザ | 65 |
| | パスワードの作成と変更 | 66 |
| | ピクチャライブラリ | 66 |
| | ピクチャの管理と参照 | 66 |
| | ピクチャの作成と変更 | 69 |
| | フレームアシスタント | 70 |
| | イベントビューワ | 73 |
| | 構成 | 73 |
| | イベントビューワを見る | 73 |
| 第3章 | ユーザモード | 75 |
| | データ読み込みとデータ書き出し | 75 |
| | 新しいエディタ | 76 |
| | 新しいオプション | 78 |
| | 新しいファイルフォーマット | 80 |
| | データ読み込み、データ書き出しの設定 | 81 |
| | ラベルエディタ | 82 |
| | メソッドの実行 | 83 |
| 第4章 | Web サーバ | 85 |
| | 接続セキュリティ | 86 |
| | パスワードオプション | 86 |
| | On Web Authentication データベースメソッド | 87 |
| | 4D Web サーバのアクセスシステム | 89 |
| | 一般 Web ユーザ | 92 |

| | |
|--------------------------------|------------|
| デフォルトHTMLルートフォルダを定義する | 93 |
| Webセッションのカスタマイズ | 95 |
| デフォルトホームページの定義 | 95 |
| データ入力管理にJavaScriptを使用 | 96 |
| 4D変数をスタティックホームページに挿入する | |
| 新しいモード | 97 |
| コンテキストを参照する新しいモード | 97 |
| 拡張ASCIIコードの直接送信 | 98 |
| 4D内の変換文字セットの変更 | 98 |
| スタティックホームページ用キャッシュ | 99 |
| HTTPリクエスト用IPアドレスの定義 | 100 |
| Webサイトに関する情報 | 101 |
| Webサーバ管理URL | 101 |
| 接続ログファイル | 102 |
| HTMLサポート | 104 |
| スタティックホームページ | 104 |
| ダイナミックページ | 104 |
| 非コンテキストモード | 107 |
| コンテキストモードと非コンテキストモード | 107 |
| あるモードから別のモードへの切り替え | 108 |
| スタートアップ時に非コンテキストモードを定義する | 110 |
| セミダイナミックページ | 111 |
| 4DACTIONタグ | 112 |
| 4DACTIONをPOSTするForm | 114 |
| On Web Connectionデータベースメソッドの変更 | 116 |
| 第5章 | |
| ランゲージ | 119 |
| 新しいコマンド | 120 |
| 配列 | 120 |
| BOOLEAN ARRAY FROM SET | 120 |
| LONGINT ARRAY FROM SELECTION | 121 |
| フォームイベント | 122 |
| SET TIMER | 122 |
| Get edited text | 124 |
| データ読み込みとデータ書き出し | 126 |
| IMPORT DATA | 126 |
| EXPORT DATA | 128 |
| 命名セクション | 130 |
| CREATE SELECTION FROM ARRAY | 130 |

| | |
|------------------------|-----|
| オブジェクトプロパティ | 131 |
| GET OBJECT RECT | 131 |
| MOVE OBJECT | 133 |
| ピクチャ | 135 |
| PICT TO GIF | 135 |
| プロセス (通信) | 137 |
| Test semaphore | 137 |
| プロセス | 138 |
| REGISTER CLIENT | 138 |
| UNREGISTER CLIENT | 141 |
| GET REGISTERED CLIENTS | 142 |
| EXECUTE ON CLIENT | 143 |
| Process aborted | 145 |
| クエリ | 146 |
| QUERY WITH ARRAY | 146 |
| Find index key | 147 |
| レコード | 148 |
| Is new record | 148 |
| Is record loaded | 149 |
| セレクション | 150 |
| HIGHLIGHT RECORDS | 150 |
| セット | 151 |
| CREATE SET FROM ARRAY | 151 |
| ストラクチャアクセス | 152 |
| SET DATABASE PARAMETER | 152 |
| Get database parameter | 154 |
| システムドキュメント | 155 |
| Select folder | 155 |
| システム環境 | 157 |
| LOG EVENT | 157 |
| Web サーバ | 158 |
| SEND HTML BLOB | 158 |
| Web Context | 161 |
| WEB CACHE STATISTICS | 162 |
| SEND HTTP REDIRECT | 163 |
| SET HTTP HEADER | 165 |
| OPEN WEB URL | 166 |
| SET HOME PAGE | 167 |
| ウインドウ | 168 |
| Open form window | 168 |
| GET FORM PROPERTIES | 171 |

| | |
|-------------------------------|------------|
| 変更されたコマンド | 172 |
| DISTINCT VALUES(配列コマンド) | 172 |
| GOTO AREA(入力制御コマンド) | 173 |
| Open document(システムドキュメントコマンド) | 174 |
| PROCESS PROPERTIES(Processes) | 175 |
| Semaphore(プロセス (通信) コマンド) | 177 |
| SEND HTML FILE(Web サーバコマンド) | 178 |
| SET INDEX(ストラクチャアクセスコマンド) | 178 |
| SET WEB TIMEOUT(Web Server) | 179 |
| フォームイベント | 180 |
| On After Keystroke | 180 |
| On Timer | 181 |
| On Resize | 181 |
| On Clicked (変更点) | 181 |
| On Double Clicked (変更点) | 182 |
| デバッガ | 183 |
| 追加情報 | 183 |
| ウインドウの管理 | 184 |
| 新しい機能 | 184 |
| トレースモードへの切り替え | 185 |
| 第6章 最適化 | 187 |
| 新しいインデックスモード | 187 |
| インデックスモードの選択 | 187 |
| どちらのモードを選ぶか? | 188 |
| データベース機能の速度増進 | 189 |
| 検索およびシーケンシャルソート | 189 |
| インデックス検索 | 189 |
| サーモメータの表示開始の遅れ | 189 |
| インデックスページをコンパクト化 | 190 |
| コマンド索引 | 193 |
| 索引 | 195 |

4th Dimension のバージョン 6.5 へようこそ。

4th Dimension と 4D Server の新しいバージョンは、デザインモード、ユーザモード、ランゲージおよび内蔵の Web サーバにおいて多くの新しい特長を提供します。

本マニュアルについて

本マニュアルは、4th Dimension と 4D Server のバージョン 6.5 における新しい特長と変更点について説明したものです。このマニュアルは下記の章から成り立っています。

デザインモード：この章ではデザインモードの新しい特長と変更点について説明します。内容は下記の通りです。

- データベースの新規作成とオープン
- ストラクチャウインドウの変更点とその使用法
- フォームエディタの変更点
- 新しいランタイムエクスプローラ
- 新しいオブジェクト検索エディタ
- コメントを書き、編集する機能
- メソッドの編集に関連したいくつかの新しい特長
- 新しいデータベースの特質
- パスワードエディタの変更点
- 新しいピクチャライブラリ

ユーザモード：この章はユーザモードの新しい特長と変更点について説明します。内容は下記の通りです。

- データの読み込みおよび書き出し機能の再設計
- ラベルエディタの変更点
- 「メソッド実行」ダイアログ内の新しいオプション

ランゲージ：この章では4Dランゲージの新しい特長と既存のコマンドの変更点について説明します。内容は下記の通りです。

新しいコマンドと変更されたコマンド

新しいフォームイベント

デバッグの強化

Webサーバ：この章では4D内蔵Webサーバの新しい特長と改良点について説明します。内容は下記の通りです。

接続のセキュリティ

カスタマイズセッション

Webサイトについての情報

HTMLサポート

非コンテキストモード

最適化：この章では4Dデータベースエンジンにおいて最適化が実施されたインデックスプロセス、シーケンシャル検索と並べ替えについて説明します。

更に、バージョン6.5には「4D Serverネットワークコンポーネント」、「4D Tools」および「Customizer Plus」に加えられた新しい特長も備わっています。これらの新しい特長については各マニュアルを参照してください。

4D 3.x.x/6.0.x データベースとの互換性

4Dのバージョン3.xおよび6.0.xで作成された4Dデータベースは、バージョン6.5と完全な互換性があります。

4D 3.x.xデータベース：バージョン6.5で開くとストラクチャファイルとデータファイルをバージョン6.5用に変換する前に2つの連続した警告を2回表示します。一旦バージョン6.5用に変換されると、バージョン3では再び開くことが出来なくなります。

4D 6.0.xデータベース：バージョン6.5で直接使用することが可能であり変換は必要ありません。しかし、一旦バージョン6.5で使用したストラクチャファイルはバージョン6.0.xで再び開くことが出来なくなります。バージョン6.0.xのストラクチャファイルを最初にバージョン6.5で使用する際、この制限についての警告が表示され注意を促します。

一方、データファイルは何も変更せずに、バージョン6.0.xおよびバージョン6.5のどちらでも使用できます。

注：変換されたデータベースのバージョン6.5の新しい機能には初期値が設定されます。これらの設定はこのマニュアル内で説明しています。

バージョン6.5のデザインモードには数多くの変更や新しい機能が加えられました。これらの新しい機能は以下の節で説明しています。

データベースの新規作成とオープン

ストラクチャウインドウの変更点とその使用法

フォーム

ランタイムエクスプローラ

オブジェクト検索エディタ

コメント

メソッドの編集

新しいデータベースプロパティ

パスワード

ピクチャライブラリ

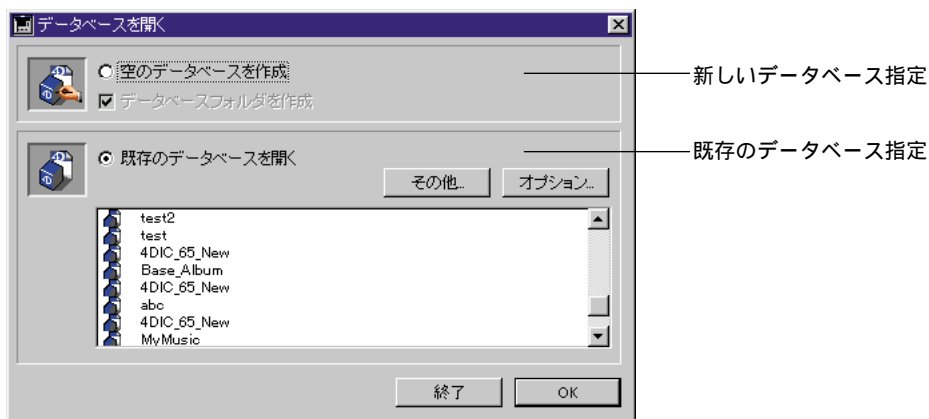
イベントビューワ (WindowsNT)

データベースの新規作成とオープン

バージョン6.5では既存データベースのオープンと新規作成は、同一ダイアログで行なうことができます。

4 D データベースの新規作成とオープンダイアログ

このダイアログにより、以下のことができます。



新規データベースの作成

既存のデータベースのオープン

追加オプションにより、ストラクチャファイルの検査、使用するデータファイルの選択もできます。

新規データベースの作成

新しくデータベースを作成するには、

1. 「空のデータベースを作成」ラジオボタンを選択する。

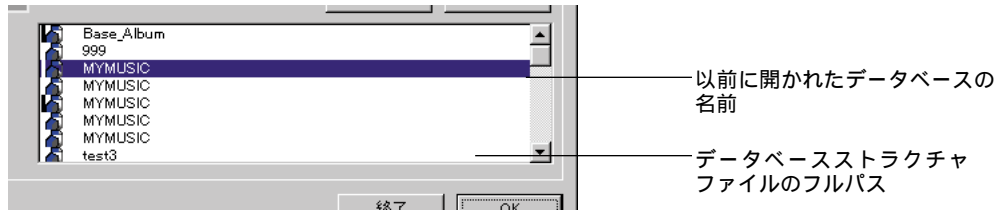


2. OKをクリックする。

標準の「ファイル保存」ダイアログが現れ、新しいストラクチャファイルの名前と作成場所を指定するとデータベースが作成されます。以前のバージョンとは異なり、バージョン6.5ではストラクチャファイルとデータファイルを同時に作成します。

既存のデータベースを開く

以前に開かれたデータベースの名前とパスを保存し、「既存のデータベースを開く」ダイアログの下部エリアに、最近開かれた順に表示します。



リストからデータベースを開くには、データベース名をダブルクリックするか、またはデータベース名を選択してOKをクリックします。

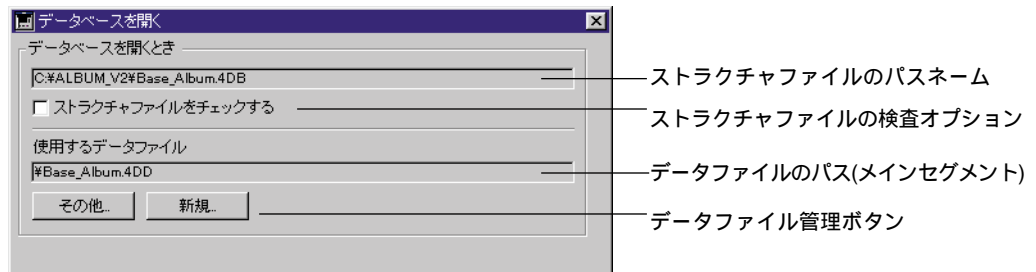
リストに表示されていないデータベースを開くには、「その他」ボタンをクリックします。標準の「ファイルオープン」ダイアログが表示され、開くことができるストラクチャファイルを選択することができます。

注：

- 一旦開いたファイルをOSレベルで削除したり、ファイルを移動または別名に変更した場合は、リストが更新され、リスト上のファイル名は削除されます。
- バージョン6.0.xまたはそれ以前の4Dで作成されたデータベースを開く方法についての詳細は、第1章の「はじめに」を参照してください。

オプション

「オプション」ボタンをクリックすると次のようなダイアログが表示され、データベースを開く際のオプションを設定することができます。



このダイアログはストラクチャファイルとデータファイル両方の名前とパスを表示します。データファイルが分割されている場合、この情報はメインセグメントのファイルに適用されます。

ストラクチャファイルをチェックする：このオプションが指定されると、データベースを開始するたびにストラクチャファイル（テーブル、フォーム、メニューバーなど）を検査します。この検査が実行されると「検査と修復」ウインドウが表示され、レポートエリアに実施されたテストの結果を表示します。

このオプションについてのより詳細は、『4D Tools リファレンスガイド』を参照してください。

その他：このボタンをクリックすると「ファイルオープン」ダイアログが表示され、現在選択されているストラクチャファイルと共に、開くことができるデータファイルを選択することができます。

新規：このボタンをクリックすると「ファイル保存」ダイアログが表示され、現在選択されているストラクチャファイルと共に、開かれる新しいデータファイルを作成することができます。

ストラクチャウインドウのインタフェースとナビゲーション

4th Dimension のストラクチャウインドウのインタフェースは変更され、操作方法が改善されました。

コンテキストメニュー

ストラクチャウインドウ内でのキーボードショートカット

テーブルの作成を確認するダイアログ

文字列フィールドの長さの表示

コンテキストメニュー

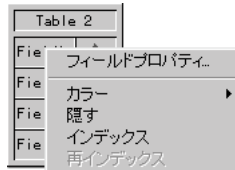
デザインモードで、コンテキストメニューを使用し、オブジェクトに対しての指示やダイアログを開くことができます。

コンテキストメニューを使用するには：

- 1 Windows ではマウスの右ボタンでオブジェクトまたはエリアをクリックする。
Macintosh ではオブジェクトまたはエリアを Control キーを押しながらクリックする。
オブジェクトまたはエリアに関連したコンテキストメニューが現れます。


ストラクチャウインドウ内のフィールドに関連したコンテキストメニュー

Windows 版



Macintosh 版



注：Macintosh では、Control キーを押すと  のようなマウスポインタが現れて、コンテキストメニューが起動されたことを示します。

2 コンテキストメニューからコマンドを選択する。

コンテキストメニュー内のコマンドは、現在のエディタ並びにクリックされたオブジェクトに応じて変わります。その名前が示すように、コンテキストメニューは、特定のコンテキストに関連した動作を実行できるようにするだけです。実行できる動作は、標準メニュー、ボタンおよびキーボードショートカットと同じものです。

ストラクチャエディタ内での選択にキーボードを使用する

キーボードショートカットを使用して、ストラクチャウインドウ内のテーブルやフィールド間を簡単に移動することができます。

次のようなショートカットが可能です。

| キー | ストラクチャエディタ内での動作 |
|---------------|--|
| | テーブルおよびサブテーブル |
| [Tab] | <ul style="list-style-type: none"> ・ストラクチャウインドウ内で次のテーブル（またはサブテーブル）を選択 ・ストラクチャウインドウが、選択されたテーブル（またはサブテーブル）の位置までスクロールし前面に表示 |
| | フィールドおよびサブフィールド |
| []と[] | 次または前のフィールドを選択 |
| [Home]と[End] | テーブル（またはサブテーブル）の最初または最後のフィールドを選択 |
| [PgUp]と[PgDn] | 次または前のフィールド「ページ」を表示 |

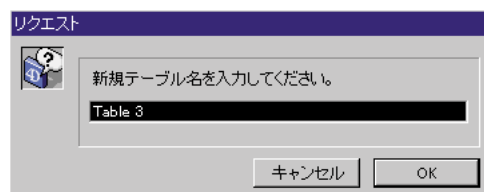
テーブルの並びの管理

作成の順序に関わりなくテーブルをクリックするか、またはTabキーを使用して前面に出すことができます。関連するテーブルの並びは保存され、指定したユーザ特有のものとなります。

テーブルの並びは、ストラクチャウインドウ内でテーブルが重なりあった場合どの順序で表示されるか、あるいはTabキーを使用した場合どの順序で選択されるかを決定します。

テーブルの作成

コンテキストメニュー内またはストラクチャメニューの「新規テーブル...」項目を選択するか、またはツールバーの対応するボタンをクリックし、新しいテーブルを作成すると下記の確認ダイアログが表示されます。



「キャンセル」ボタンをクリックするとテーブルは作成されず、ストラクチャは変更されません。

文字列フィールドの長さの表示

ストラクチャウインドウでは、文字列フィールドに定義された文字数が表示されます。

| | |
|---------|-----|
| Table 2 | |
| 顧客ID | L |
| 会社名 | A71 |
| 担当者 | A41 |

文字列フィールド内の文字数

フォーム

バージョン6.5では、フォームの作成や変更をする際に、数多くの変更や新しい機能が加えられました。これらの新しい機能は下記の項で説明しています。

デフォルトフォームの作成：フォーム自動作成機能をオン・オフすることができます。

フォームエディタの変更：フォームエディタのインタフェースおよびツール（メニュー、パレット、表示フィルタ、マーカーなど）が変更されました。

オブジェクトのプロパティリスト：この新しいパレットで、フォーム内のオブジェクトプロパティをダイナミックリストとして、参照や変更ができます。

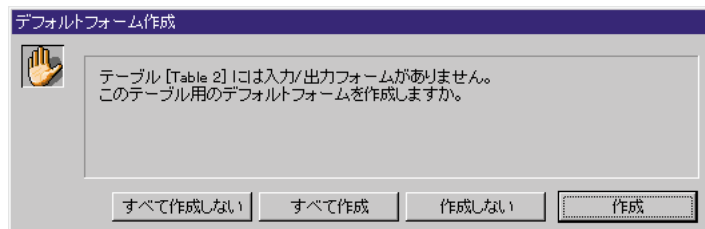
フォームオブジェクト：スプリッタと呼ばれるフォームオブジェクトの新しいタイプが使用できます。フォーム内に直接テーブル名やフィールド名を置くことができます。また、既存のオブジェクトに対する新しい可能性もあります。ダイナミックオブジェクトのバックグラウンドの透過指定、ピクチャボタンの新しいモードなどです。

リストフォームのヘッダ：アクティブオブジェクトをリストフォーム（出力フォーム）のヘッダエリアに置くことができます。

オンラインヘルプ：オンラインヘルプをデータベースと関連させることができ、それぞれのフォームに「トピック」を関連させることもできます。

デフォルトフォームの作成

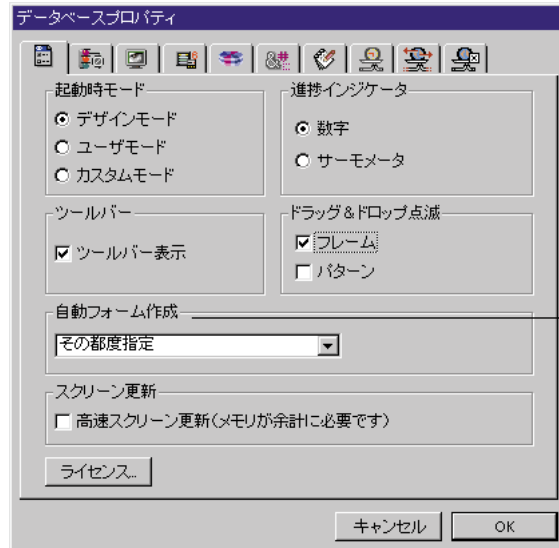
デザインモード内でテーブルを作成し、ユーザモードに移行する時、新しいテーブル用にフォームが作成されていないと、4Dはデフォルトの入力および出力フォームを作成するかどうかを聞いてきます。



この警告を表示させないことができます。

デフォルトフォームの自動作成方法を指示するには、

- 1 「データベースプロパティ」ダイアログ内で、「一般」タブをクリックする。



自動フォーム作成オプション

- 「自動フォーム作成」メニューには、3つのオプションがあります。

常に作成しない：「警告」ダイアログは表示されず、デフォルトフォームは作成されません。

その都度指定：テーブル用のフォームが何も作成されていない場合に、「警告」ダイアログを表示します（デフォルトオプション）。

常にすべて作成する：「警告」ダイアログは表示されず、すべてのテーブル用にデフォルトフォームが自動的に作成されます。

- 2 いずれかを選択し、「OK」ボタンをクリックする。

この設定は、設定した4Dで開く全データベースに適用されます。

フォームエディタの変更

フォームエディタのインタフェースは、フォームの作成と変更をより簡単に行なえるように修正されました。

コンテキストメニュー

選択範囲内に完全に含まれるオブジェクト選択ショートカットの修正

メニューバーのフォームへの関連付けの修正

フォームエディタ内でのエレメント表示/非表示ができる新しいオプション

マーカーラベル

フォームの境界表示

フォームメソッドへの直接アクセス

パレットのフォーカス自動管理

新しいツールパレット

フォームのページを管理する新しいコマンド
フォームの配置先の選択
アライメントアシスタント
「オブジェクト複製」ダイアログのマトリクス

コンテキストメニュー

フォームエディタでコンテキストメニューを使うことができます。コンテキストメニューを表示するには、Windowsではマウスの右ボタンをクリックし、MacintoshではControlキーを押しながらクリックします。

常に使用可能な標準の編集コマンド（コピー、ペーストなど）とは別に、フォームエディタのコンテキストメニューの内容は、どこをクリックするかによって変わります。

オブジェクトをクリックすると、メニューはそのオブジェクトに関連したオプションとコマンド（オブジェクトプロパティ、オブジェクトメソッド、カラー、整列、境界線スタイル、自動サイズ、自動アクション(オブジェクトによる)およびレベル）が選択できます。オブジェクトによっては追加コマンドも選択できます。例えば、フィールドを追加コマンドはオブジェクトがフィールドである場合、リストコマンドはオブジェクトが階層リストである場合に現れます。

オブジェクトをクリックせずにフォームエディタのウインドウをクリックすると、メニューがそのフォームに関連したオプションとコマンド（フォームプロパティ、フォームメソッド、グリッドあり、表示（詳細は、この章の「フォームエディタ内のエレメント表示/非表示」を参照してください）およびページ移動を提供します。

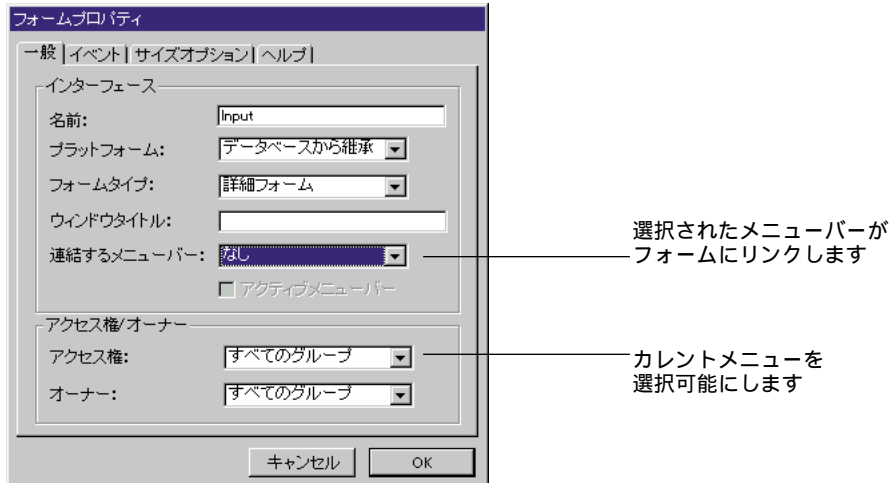
選択範囲内に完全に含まれるオブジェクト選択ショートカット

Macintoshでは、Controlキーを押しながらクリックすることでコンテキストメニューを表示するようになりました。しかし、以前のバージョンのフォームエディタでは、このショートカットを選択範囲内に完全に含まれるオブジェクトの選択に使用していました。

そこで、新しいバージョンではこのショートカットを、optionキーを押しながらクリック（Macintoshの場合）またはAltキーを押しながらクリック（Windowsの場合）に変更しました。

メニューバーをフォームに関連付ける

フォームメニュー内の「メニューバー連結」は削除されました。この機能はフォームプロパティで設定することができます。



このポップアップにはデータベース内で作成されたすべてのメニューバーが含まれています。フォームに関連させたいメニューバーを選択することができます。

連結するメニューバーオプションが選択されると、4Dはカスタムモードでカレントメニューバーの右側にフォームに、関連したメニューバーを表示します。これまでの4Dのバージョンでは、関連するメニューバーの番号の前に「-」（マイナス）記号を付けることによって指定できました。詳細は『4th Dimension デザインリファレンス』マニュアルを参照してください。

フォームエディタ内のエレメント表示/非表示

フォームエディタ内のほとんどのインタフェースエレメントを表示/非表示にすることができます。この機能により、フォーム内で作成したいエレメントだけ、または見たいエレメントだけを見せることや、使いたいツールのみを見せることができます。この設定はフォームエディタで表示するフォームすべてに適用されます。

1 フォームメニューから表示を選択する。

または、

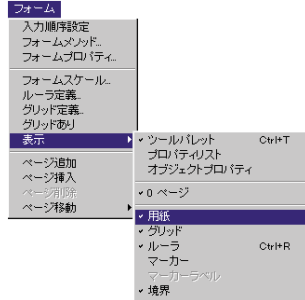
フォームエディタ内に現れるコンテキストメニューの表示コマンドを使用する。

Windows では、オブジェクト上以外でマウスの右ボタンをクリックする。

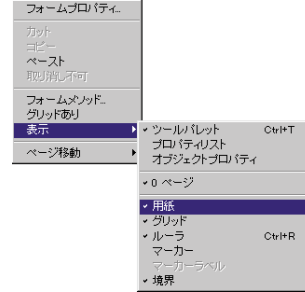
Macintosh では、オブジェクト上以外で Control キーを押しながらクリックをする。

設定可能なエレメントリストのサブメニューが表示されます。

フォームメニュー



コンテキスト
メニュー



各エレメントのチェックマークは、そのエレメントが表示されることを示します。エレメントを隠すには、隠したいエレメントを選択してチェックマークを外します。

2 対象エレメントを選択する。

このメニューのコマンドについて説明します。

ツールパレット：フォームエディタのツールパレットの表示/非表示。

このパレットは以前のバージョンのツールやオブジェクトパレットを再グループ化しています。詳細は、この章の「新しいツールパレット」を参照してください。

プロパティリスト：プロパティリストの表示/非表示。

このリストはバージョン6.5の新しい機能であり、「プロパティリスト」で説明しています。プロパティリストは、オブジェクトプロパティに代わるものです。プロパティリストとオブジェクトプロパティの間で切り替えることができ、1つが選択されている時には、もう一方は選択されません。

オブジェクトプロパティ：オブジェクトプロパティの表示/非表示。

このパレットはプロパティリストに置き換えることができます（上記参照）。パレットのフォーカスの管理方法がバージョン6.5では修正されています。詳細は、この章の「パレットのフォーカス自動管理」を参照してください。

0ページ：0ページオブジェクトの表示/非表示。

このオプションを使うとフォームの現在のページ内のオブジェクトと0ページ内のオブジェクトを区別できます。

用紙：グレーの線として表示されている印刷ページの境界表示/非表示。

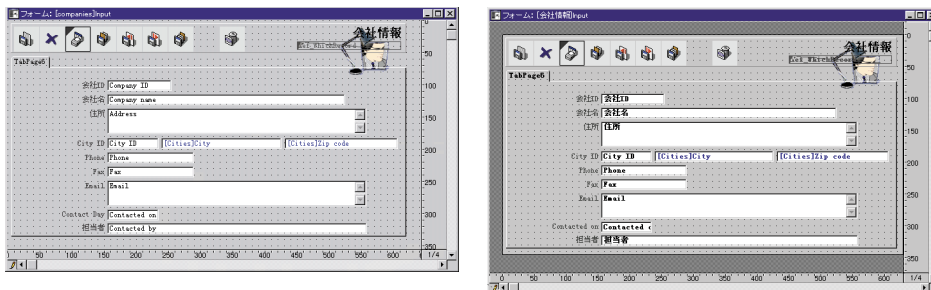
このオプションは、境界（下記参照）オプションが選択されていると、顕著な効果はありません。フォームのサイズが印刷ページよりも小さいと、ページの境界はフォームのビューエリアの外側に置かれるため、エリア内では見えません。

shiftキーを押しながらクリックし、ドラッグすると、その下の全マーカーを同時に移動することができます。

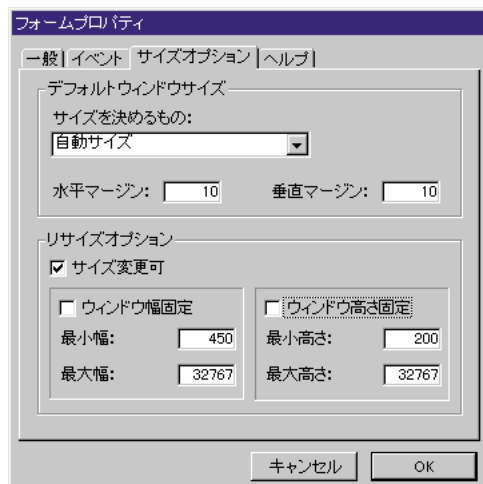
フォームの境界表示

この新しいオプションは、フォームをユーザモードでどのように表示されるか、フォームエディタで見ることができます。この機能は結果を見るためにユーザモードへ移動することなくフォームを修正することができるため、非常に便利な機能です。

フォームの境界を見るには、フォームメニューの表示から境界を選択します（詳細は、この章の「フォームエディタ内のエレメント表示/非表示」を参照してください）。



フォームのデフォルトウィンドウサイズは、フォームプロパティの「サイズオプション」ページに定義されている値から計算されます。「サイズを決めるもの」、「水平マージン」、「垂直マージン」オプションの設定がフォームの境界に影響します。



注：コンテキストメニュー（フォームエディタ内のオブジェクトでない部分を、Windowsではマウスの右ボタン、MacintoshではControlキーを押しながらクリック）を使ってフォームプロパティを表示することができます。

デフォルトでは、「サイズを決めるもの」は「自動サイズ」が選択されます。これは、フォームの境界がフォーム上のオブジェクトに応じて計算されることを意味します。フォームの境界近くに置かれたオブジェクトを移動、またはサイズ変更すると、ウィンドウのサイズがオブジェクトに合うように修正されます。

フォームメソッドへの直接アクセス

下記の3通りのいずれかで、フォームエディタから直接フォームメソッドを表示することができます。

フォームメニューからフォームメソッドを選択する。

フォームエディタのコンテキストメニュー（フォームエディタ内のオブジェクト以外の部分を、Windowsではマウスの右ボタン、MacintoshではControlキーを押しながらクリック）を表示させ、フォームメソッドを選択する。

プロパティリスト（「プロパティリスト」を参照）が開かれている間に、フォームエディタ内のオブジェクト以外の部分をクリックする、またはリストのトップに位置しているドロップダウンメニュー内のフォームからフォーム名を選択した後、フォームメソッド行の隣の「編集...」ボタンをクリックし、フォームメソッドを開く。

パレットのフォーカス自動管理

オブジェクトとフローティングウィンドウの間のフォーカス管理は自動で行ないます。標準のコピー＆ペースト（貼り付け）やキーボードショートカットはフローティングウィンドウに対してではなく、「フォーカス」を持つエレメントに対して実行されます。

以前のバージョン（特にオブジェクトプロパティパレット内）では、スイッチを使い、手動でフォーカスの指定をしなければなりません。バージョン6.5ではスイッチは取り除かれました。

バージョン6.5では、ユーザが最後にクリックしたものが自動的にフォーカスを持つエレメント（フォームエディタ内のオブジェクトまたはフローティングパレット）となります。

この原則はフローティングパレットを持つすべての4Dのエディタに適用されます。しかし、パレット内にフォーカスするオブジェクトがない場合、パレットにフォーカスが移ることはありません。

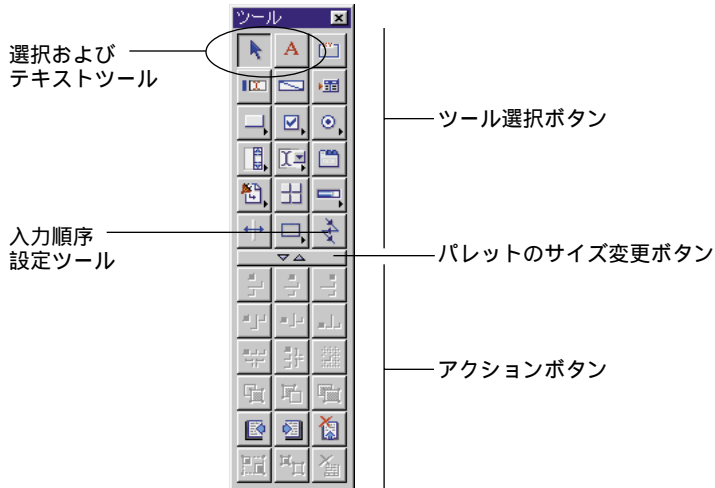
新しいツールパレット

4D 6.0.xオブジェクトパレットとツールパレットは、新しいツールパレットに統合されました。

新しいツールパレットとしての機能は同じです。

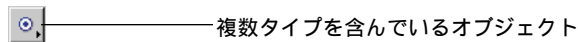
パレットの上部エリアはツールエリアで、再グループ化してあります。オブジェクトを作成するには、ツールを選択し、ドラッグしてフォーム編集エリアにドロップします。デフォルト値が設定されたオブジェクトがフォーム内に挿入され、選択されます。

オブジェクトの自動作成でない場合は、オブジェクトプロパティパレット (またはプロパティリスト) が現れます。



各オブジェクト (ボタン、ラジオボタン、メニュー、...)は、タイプごとにツールパレット内でグループ化されています。各オブジェクトタイプには異なったタイプが複数含まれています。現在ツールパレットに表示されているタイプがフォーム内に挿入されるオブジェクトタイプです。

複数のタイプを含んでいるオブジェクトには、ボタンの右下の隅に小さな三角形があります。



オブジェクトのタイプを選択するには、

- 1 「オブジェクト」ボタンをクリックし、そのままマウスボタンを押す。

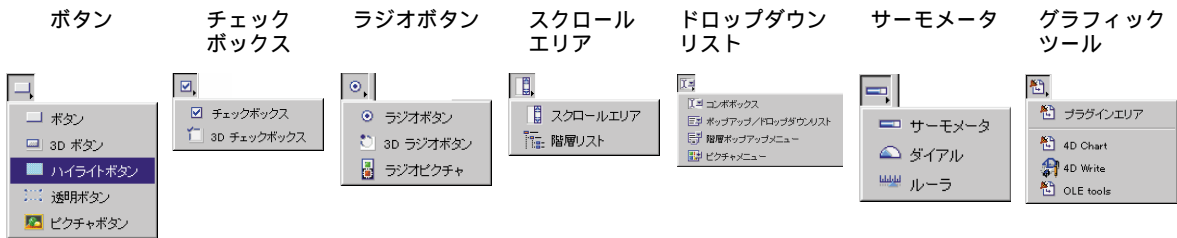
または

Windows では、マウスの右ボタンで「オブジェクト」ボタンをクリックする。

Macintosh では、Control キーを押しながら「オブジェクト」ボタンをクリックする。


表示されたメニューで選択されたオブジェクトタイプが現在のタイプになります。

ツールパレット内で選択できる各オブジェクトタイプの種類を示します。




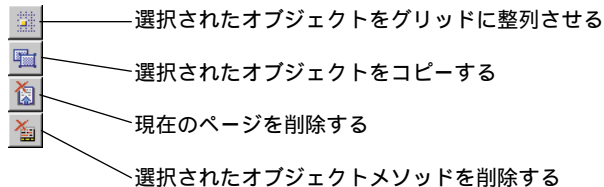
注：

- ・プラグインエリアボタンも、データベース中にインストールされたプラグインに応じて変化します。
- ・変数の入力不可タイプの指定は以前のように行わず、変数タイプオブジェクト用のプロパティ（オブジェクトプロパティまたはプロパティリストで指定可能）で設定する「入力可」の指定で行ないます。

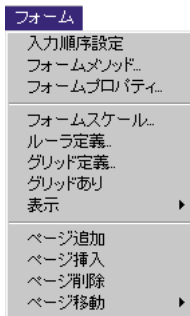
パレットの下部エリアはアクションエリアで、再グループ化してあります。パレットの中央部にある  ボタンをクリックすると、このエリアを隠すことができ、ツールエリアのみが表示されます。



アクションエリアを表示するには、 ボタンをもう一度クリックします。アクションエリア内の動作は、「4つの新しい動作」以外は以前のバージョンでのものと同一です。詳細は『4th Dimension デザインリファレンス』を参照してください。



新しいページ管理コマンド



フォームメニュー内のフォームページ管理機能を実行できます。

ページ追加：最後のページの後に1ページ加えます。

ページ挿入：現在のページの前に1ページ挿入します。

ページ削除：現在のページを削除します。

ページ移動：指定されたページに移動します。

注：ページ移動コマンドは、他の2つの方法でも実行することができます。

- ・ルーラが表示されている時、フォームエディタの右下にあるポップアップメニューを使用する。
- ・フォームエディタのコンテキストメニュー（フォームエディタ内のオブジェクト以外の部分を、Windowsではマウスの右ボタン、MacintoshではControlキーを押しながらクリック）を使用する。

フォームタイプの選択

入力フォームの
リスト



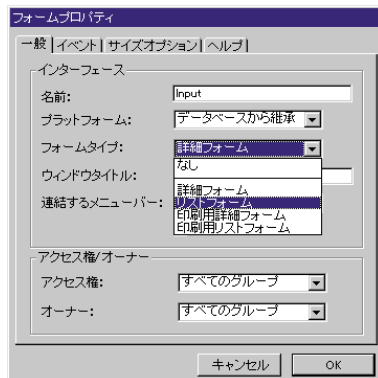
出力フォームの
リスト

バージョン6.5では、フォームタイプ（詳細フォーム、リストフォーム、印刷用詳細フォームまたは印刷用リストフォーム）はフォームプロパティで修正できるようになりました。以前のバージョンでは、フォームを作成した時を除いて定義できませんでした。

バージョン6.5では、フォームタイプを選択することで、入力または出力フォームの選択リスト（ユーザモードでのテーブルリストウインドウ）に対応するフォームのみを表示させることができます。

フォームタイプを変更するには、

- 1 フォームプロパティ内のフォームタイプポップアップから選択する。



- 2 「OK」ボタンをクリックする。

フォームタイプが「なし」の場合は、テーブルリストウインドウ内の出力および入力フォームメニューの両方に表示されます。以前のバージョンで作成され、バージョン6.5で開かれたデータベースでは、すべてのフォームに「なし」フォームタイプが割り当てられます。

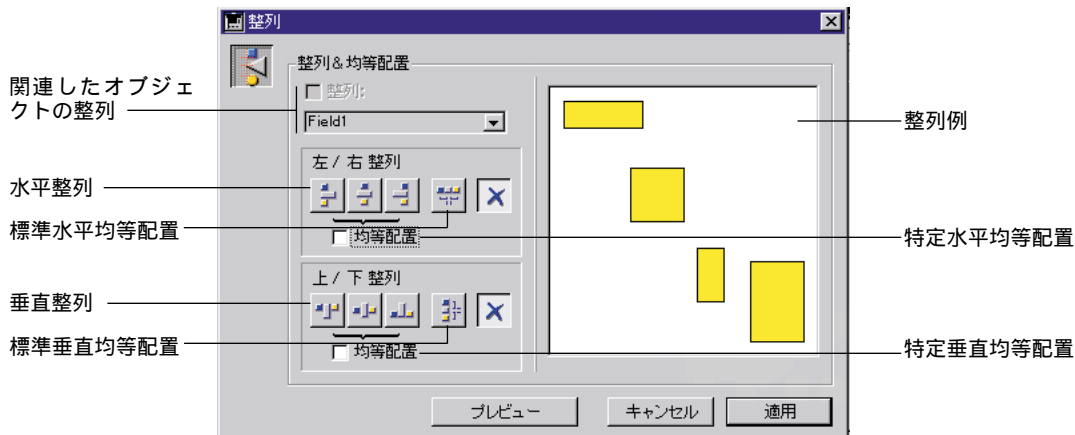
整列アシスタント

以前のバージョンのように、ツールパレット内のボタンを使用してフォームオブジェクトの整列と均等配置ができます。しかし、バージョン6.5ではオブジェクトメニュー内およびフォームエディタのコンテキストメニュー内の整列サブメニューを使用することもできます。

アシスタントを使用すれば、選択されたオブジェクトを異なる方法で整列し、均等配置することも可能です。このアシスタントは、関連したオブジェクトを整列することや、適用する前に設定の結果をプレビューすることもできます。

整列アシスタントを表示するには、整列または均等配置させるオブジェクトを選択して、整列サブメニュー内の「整列」を選びます。

次のウィンドウが現れます。



整列

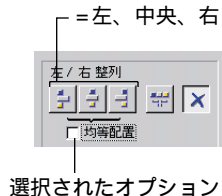
- ・選択したオブジェクトを整列させるには、アイコン上をクリックします。
- ・整列オプションをチェックし、リストから参照オブジェクト名を選択することで、参照オブジェクトを基点とした整列を行うことができます（この場合、参照オブジェクトの位置は変化しません）。

均等配置

- ・選択されたオブジェクトを均等に配置するには、水平または垂直の均等配置アイコンをクリックします。各オブジェクトのサイドが等間隔で離れるように配置されます（標準均等配置）。
- ・特定均等配置も適応できます。例えば、各オブジェクトの右サイド（外側の枠ではない）が等間隔で離れるように均等配置することができます。その場合は、特定均等配置オプションをチェックしてください。このオプションがチェックされると、その上のアイコンが均等配置する際に適用されます。

水平均等配置では、各アイコンは次の均等配置に対応します：選択されたオブジェクトの左側、中央（垂直）、右側が等間隔になるように配置します。

垂直均等配置では各アイコンは次の均等配置に対応します：選択されたオブジェクトの上辺、中央（水平）、下辺が等間隔になるように配置します。



整列例は、指定された並び方の例を図示します。また、「プレビュー」ボタンをクリックすることによって、指定した並び方の結果をプレビューすることもできます。プレビューはダイアログがそのままの状態でもフォームに反映され、適用またはキャンセルを選択できます。

マトリックス上でオブジェクトを複製

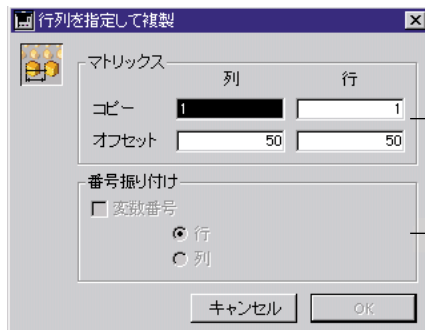
バージョン6.5のフォームエディタには、オブジェクトを複製することができる新しいダイアログがあります。

1回の操作で、1つのオブジェクトまたは複数のオブジェクトを、必要なだけコピーすることができます。また、各複製されたオブジェクト間のオフセットも定義することができます。

このダイアログにより、自動的に番号付けされた変数の並びも作成することができます。

複数のオブジェクトを複製するには、

- 1 複製するためのオブジェクトを選択する。
- 2 オブジェクトメニューから「行列を指定して複製...」を選択する。
「行列を指定して複製」ダイアログが現れます。



複数行列の定義付け

番号付けされた変数の並びを作成するためのオプション（エリアがアクティブになるのは変数が選択された場合のみ）

- 3 上部エリアには、作成したいオブジェクトの列数と行数を入力する。
例えば、3列と2行のオブジェクトが欲しい場合、列エリアに3、行エリアに2を入力します。
オブジェクトの水平方向に3つのコピーが欲しい場合には、列エリアに3を入力し、行エリアはデフォルト値1をそのまま残します。
- 4 行および列に、各コピーオブジェクト間のオフセットをポイント値で定義する。
これは各コピーオブジェクトに適用されます。
例えば、高さが50ポイントのオブジェクトを、20ポイントの間隔を空けて垂直方向にコピーしたい場合は、列の「オフセット」エリアに70を入力します。
- 5 変数の並びを作成したい場合には、「番号振り付け」オプションを選択する（選択しない場合はステップ7へ）。
このオプションは、選択されたオブジェクトが変数である場合のみ選択可になります。
- 6 変数が番号付けされるべき方向を、行または列によって選択する。

7「OK」ボタンをクリックする。

指定された数のオブジェクトの列および行が作成されます。

注：「グリッドにオブジェクトを作成」コマンドでも、グラフィカルな配列を使用し、自動的に番号付けされた変数の並びを作成することができます。この機能は、以前のバージョンでも存在していて「グリッドで変数作成」と呼ばれていました。詳細は、『4th Dimension デザインリファレンス』を参照してください。

プロパティリスト

バージョン6.5では、オブジェクトプロパティパレットの代用としてプロパティリストがあります。

プロパティリストとオブジェクトプロパティパレット

これら2つのツールは、フォームオブジェクトを管理するために同じ機能を提供します。しかし、プロパティリストは下記のエリアで異なります。

プロパティリストの内容は選択されたオブジェクトを反映します。オブジェクトとそのタイプに応じて可能なテーマとプロパティのみが表示されます。さらに、リストの内容は設定を変更するたびに変化します。例えば、フィールドにフォーカス属性を割り当てると、タブ有効属性が選択可能となります。変数を「入力可」として定義すると、入力制御テーマが現れます。

プロパティリストは、フォームプロパティを表示させることができます（詳細は、この章の「プロパティリスト内にフォームプロパティを表示させる」を参照してください）。

プロパティリストは、同じファミリに属する別のオブジェクトタイプに変更することができます（詳細は、この章の「オブジェクトタイプの変更」を参照してください）。

プロパティリストは、フォーム上の変数をタイプ分け（実数、数値、...）することができます（詳細は、この章の「フォーム内の変数のタイプ分け」を参照してください）。

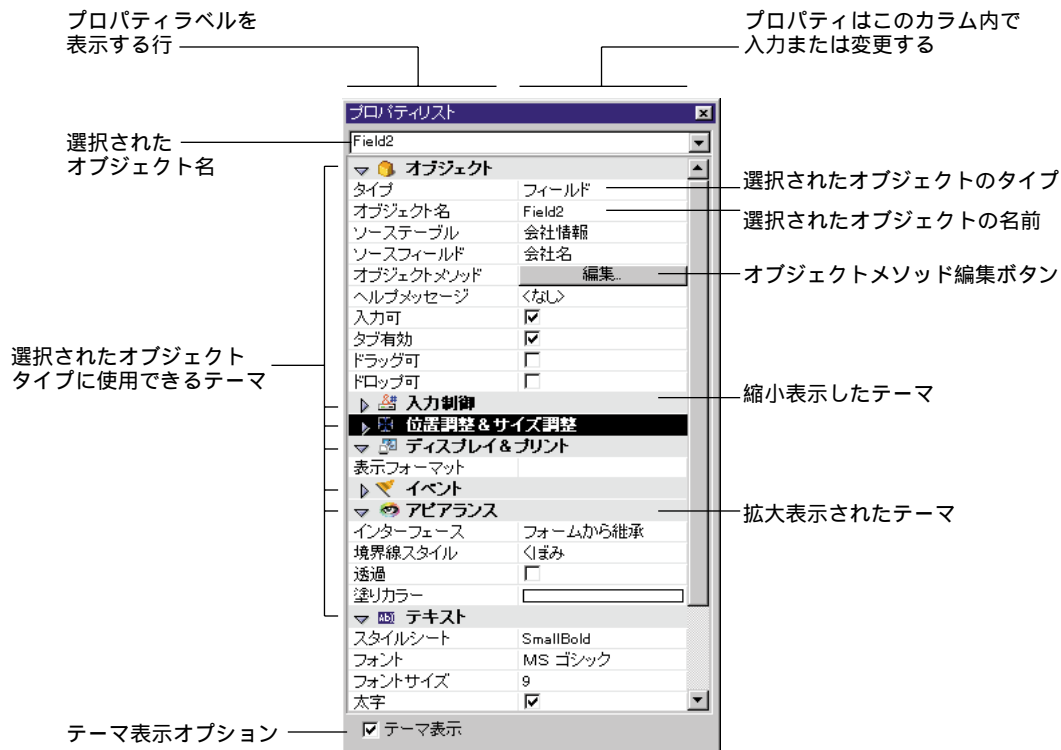
プロパティリストは、スタイルシートやヒント(Tips)を変更することはできません。

注：全体としてプロパティの種類とその値は、以前のバージョンのものと同一です。本マニュアルでは、プロパティリストの新しい機能のみを記述していますので、詳細は『4th Dimension デザインリファレンス』を参照してください。

プロパティリストの表示

2つのパレットを同時に表示させることはできませんが、必要性和作業の性格に応じてどちらを使用するかを選択することができます。プロパティリストを表示するには、フォームメニュー内の表示サブメニューから、あるいはフォームエディタのコンテキストメニュー（フォームエディタ内のオブジェクト以外の部分を、Windowsではマウスの右ボタン、MacintoshではControlキーを押しながらクリック）の表示サブメニューから選択します。

プロパティリストは下記のようにになっています。



注：オブジェクトプロパティパレット内のページとプロパティリスト内のテーマは正確に対応していません。

プロパティリストの使用

プロパティリストの主な機能を紹介します。

ウインドウの下部右端をクリックすることによってウインドウのサイズを変更できます。

それぞれのテーマは、矢印をクリックすることによって拡大/縮小することができ、作業したいプロパティのみを表示させることができます。「テーマ表示オプション」のチェックを外すことによって、テーマの名前を隠すこともできます。この場合、プロパティは連続したリストとして表示されます。

タイプに関わりなく複数のオブジェクトを選択すると、共通のテーマやプロパティのみが表示されます。

左側は使用可能なプロパティとテーマを、右側は各プロパティの現在値を表わします。プロパティを選択して新しい値を設定することができます。各プロパティは、コンボボックス、チェックボックス、ピクチャメニュー、ボタン、ドロップダウンメニュー、その他などで設定します。

| タイプ | フィールド | タイプ | フィールド |
|------------|-------------------------------------|------------|----------|
| オブジェクト名 | Field2 | オブジェクト名 | Field2 |
| ソーステーブル | 会社情報 | ソーステーブル | 会社情報 |
| ソースフィールド | 会社名 | ソースフィールド | Contacts |
| オブジェクトメソッド | 編集... | オブジェクトメソッド | 会社情報 |
| ヘルプメッセージ | くなし | ヘルプメッセージ | 製品情報 |
| 入力可 | <input checked="" type="checkbox"/> | 入力可 | 発注書 |
| タブ有効 | <input checked="" type="checkbox"/> | ヘルプメッセージ | 発注明細 |
| ドラッグ可 | <input type="checkbox"/> | | |
| ドロップ可 | <input type="checkbox"/> | | |

プロパティリストでは、次のキーボードショートカットが使用できます。

矢印キー：セルからセルへの移動

PgUp および PgDn：表示されているリスト内の最初と最後のセルへの移動

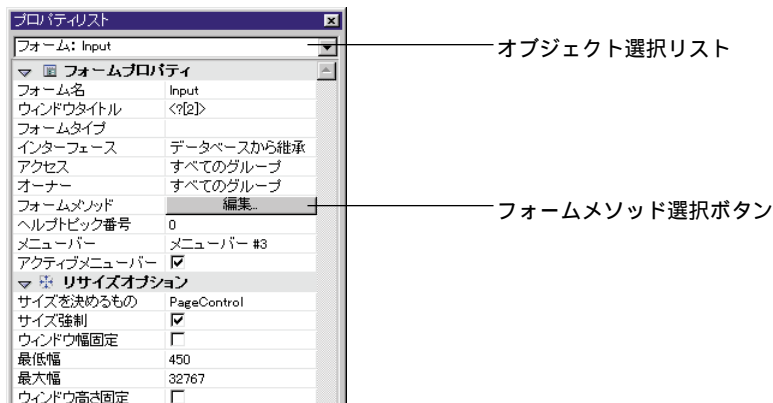
Home および End：リスト中の最初と最後のセルへの移動

イベントタイトルでの Ctrl キーを押しながらクリック（Windows の場合）または command キーを押しながらクリック（Macintosh の場合）：クリックしたイベントの初期状態に応じて、すべてのイベントを選択または非選択にする

テーマ名上での Ctrl を押しながらクリック（Windows の場合）または command を押しながらクリック（Macintosh の場合）：すべてのテーマを拡大/縮小する

プロパティリスト内にフォームプロパティを表示させる

フォームエディタウインドウ内の空白エリアでクリックすると、プロパティリストはフォームプロパティを表示します。また、プロパティリストウインドウのトップにあるオブジェクト選択リストのフォームからフォーム名を選択しても、このリストを表示できます。



プロパティリストウインドウは、フォーム属性の表示やフォームメソッドの編集を可能にします。

オブジェクトタイプの変更

プロパティリストは、アクティブ、非アクティブに関わらず、すべてのオブジェクトタイプを他のオブジェクトタイプに変更することを可能にします。また、フィールドを変数に変更することも、その逆もできます。

プロパティリストは、論理的なタイプ変更のみを許しています（例えば、チェックボックスをサーモメータに変更するのは意味をもちませんが、ラジオボタンに変更するのは可能です）。各オブジェクトタイプは「ファミリー」に属し、同じファミリー内の別のオブジェクトにのみ変更できます。

オブジェクトタイプを変更するには、

- 1 変更したいフォーム内のオブジェクトを選択する。
- 2 プロパティリスト内で、タイプメニューを引き出す。
同じファミリーのオブジェクトタイプのみがリストに表示されます。

スクロール可能な
エリアのタイプ

| ▼ オブジェクト | |
|----------|----------|
| タイプ | スクロールエリア |
| オブジェクト名 | スクロールエリア |
| 変数名 | 階層リスト |
| 変数タイプ | 文字 |

フィールド用の
タイプ

| ▼ オブジェクト | |
|----------|-------|
| タイプ | フィールド |
| オブジェクト名 | フィールド |
| ソーステーブル | 変数 |
| ソースフィールド | 住所 |

- 3 新しいオブジェクトタイプを選択する。

リストは新しいオブジェクトタイプのプロパティを表示するために再構成されます。オブジェクトの名前、プロパティ（入力可、サイズ、色など）およびメソッドがあれば保存されます。

注：変数をフィールドに変更する場合、最初のテーブルの最初のフィールドをデフォルトでオブジェクトに割り当てるため、ソーステーブルおよびソースフィールド内に変更後のテーブルとフィールドを手動で指定してください。

フォーム内の変数のタイプ分け

プロパティリストは、フォーム内の変数の「タイプ分け」の概念も取り入れています。

フォーム内の変数のタイプを定義するには、変数タイプでタイプを選択しなければなりません。

| ▼ オブジェクト | |
|------------|--------------------------|
| タイプ | 変数 |
| オブジェクト名 | 変数1 |
| 変数名 | 変数1 |
| 変数タイプ | 文字 |
| オブジェクトメソッド | 文字 |
| ヘルプメッセージ | テキスト |
| 入力可 | 数値 |
| タブ有効 | 日付 |
| ドラッグ可 | 時間 |
| ドロップ可 | <input type="checkbox"/> |

変数のタイプ

デフォルトでは、変数は実数タイプになります。定義されたタイプの適切な表示/印刷フォーマットだけが表示され、正しいフォーマットを見つけるのが容易になりました。

「タイプ分け」は表示フォーマットの管理にのみ使用され、4D言語からは独立したものです。メソッド内での変数宣言に代わるものではありません。

フォームオブジェクト

バージョン6.5は、フォームオブジェクトについて数多くの新しい機能と管理方法を取り入れています。

新オブジェクトのスプリッタ

ダイナミックテーブル名またはダイナミックフィールド名の挿入

Tips内のダイナミックリファレンス

ドラッグ&ドロップによるリストや階層メニューの作成

オブジェクトのバックグラウンドの管理

カラーラベル

ブラットフォームのインタフェースの変更

フォーカス可能と入力可能の関係の変更

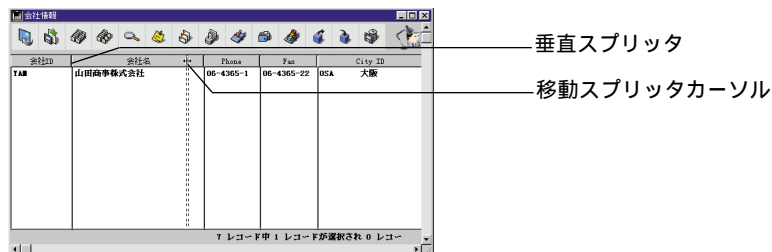
ピクチャボタンのアニメーション

スプリッタ

スプリッタは、バージョン6.5の新しいフォームオブジェクトのタイプです。1つのフォームを2つのエリアに分割し、ユーザがスプリッタを移動させることにより、分割したエリアの拡大/縮小が可能になります。水平でも垂直でも設定できます。

スプリッタは、各オブジェクトのサイズ変更プロパティを考慮に入れており、ユーザがデータベースのインタフェースを完全にカスタマイズできるようにしています。

スプリッタは、通常カラムのサイズが変更できるように出力フォーム（カスタムモード）で使用されます。



注：バージョン6.5で作成された出力フォーム（少なくとも2つのフィールドを含んでいる場合）は、自動的に垂直スプリッタが設定されます（カスタムモードで）。

スプリッタの一般的な機能を紹介します。


1つのフォーム内には好きなだけスプリッタを置くことができ、同じフォーム内で水平および垂直スプリッタを同時に使用することもできます。

スプリッタはオブジェクトをカットすることができます。このオブジェクトは、スプリッタが移動された時にサイズ変更されます。

スプリッタは、ユーザモードの出力フォーム内では使用できません。

0ページにスプリッタを置くことはできません。

スプリッタを使ってフォームをサイズ変更すると、フォームの新しいサイズはフォームが表示されている間のみ有効です。フォームが閉じられると、初期値のサイズが再度保存されます。

スプリッタ
ボタン 

スプリッタをフォーム内に挿入するには、オブジェクトパレット内のスプリッタボタンを使用します。

設定されると線として、スプリッタが表示されます。線種や色など境界線のスタイルを変更することもできます。

必要に応じてスプリッタをフォーム上に置いてください。

隣接するオブジェクトプロパティとの相互作用

スプリッタはその周囲にあるオブジェクトと、これらオブジェクトのサイズ変更オプションに応じて相互作用します。

| オブジェクトのサイズ変更オプション | 水平スプリッタ上、または垂直スプリッタ ¹ の左側のオブジェクト | 水平スプリッタの下、または垂直スプリッタ ² の右側のオブジェクト |
|-------------------|---|--|
| なし | そのまま残ります | スプリッタと共に移動し、スプリッタからの位置は変わりません。 |
| サイズ変更 | 元の位置を保ちながらスプリッタの新しい位置に応じてサイズが変更します。 | |
| 移動 | スプリッタと共に移動します。 | |

1.この位置にあるオブジェクトにサイズ変更オプションが何も定義されていなければ、水平スプリッタを上に移動させる場合や垂直スプリッタを左に移動させる場合の限界店として使用されます。

2.水平スプリッタを下に移動させる場合や垂直スプリッタを右に移動させる場合の終了地点は、ウインドウの境界または他のスプリッタのいずれかになります。この終了地点は、移動されたオブジェクトがフォーム内で完全に見えており、他のスプリッタ領域へ侵入しないような位置が計算されます。

注：スプリッタを定義する長方形内に完全に含まれるオブジェクトは、スプリッタと同時に移動されます。

スプリッタのメソッド管理

オブジェクトメソッドをスプリッタに関連付けることができ、ユーザがスプリッタの移動を終えた時に実行されます。

各スプリッタは倍長整数タイプの変数として設定および参照することができ、オブジェクトメソッドやフォームメソッドで使うことができます。値はピクセルで、スプリッタの移動距離を表わします。

値が負の場合、スプリッタは上方へ、または左側へ移動されました。

値が正の場合、スプリッタは下方へ、または右側へ移動されました。

値が0の場合、スプリッタは元の位置です。

スプリッタの値を変更するだけでメソッドからスプリッタの移動を制御することも可能です。例えば、split1 という名前の垂直スプリッタに「split1:=-10」の命令を実行すると、split1 スプリッタは10ピクセル左側に移動します。

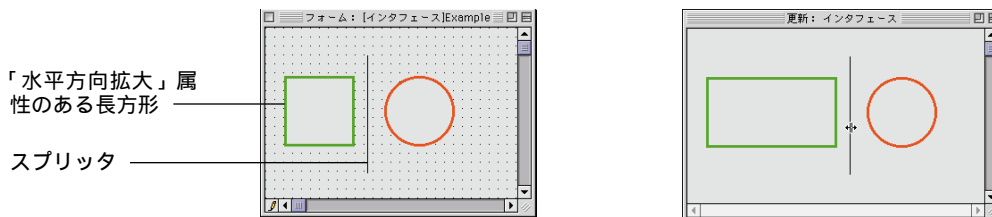
実際に移動するのは、命令を含んだフォームメソッドやオブジェクトメソッドの実行後に行われます。

自動スプリッタ

自動動作のオブジェクトメソッドを持つ透明ボタンを使用することで、独自のスプリッタインタフェースを作ることができます。

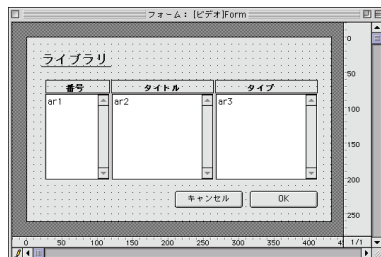
例

例1：入力フォームに長方形、垂直スプリッタおよび円があるとします。長方形には「水平方向拡大」属性が設定されています。スプリッタが右または左に移動されると、長方形は大きくなったり小さくなったりしますが、円は移動してもサイズは変更されません。ウインドウがサイズ変更されても、オブジェクトは位置やサイズを変更しません。

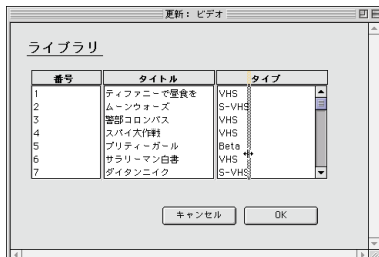


例2：入力フォームに「水平方向拡大」属性が設定されている3つのグループ化された配列があるとします。またこれらのタイトルと垂直スプリッタが各カラムの間に置かれています。このように各カラムの関連サイズを変更することができます。

デザインモード



ユーザモード



ダイナミックテーブル名またはダイナミックフィールド名の挿入

テーブル名やフィールド名を直接フォーム内にテキストとして挿入することができます。

テーブル名やフィールド名を直接フォーム内に置くと、データベース全体を通じてテーブル名やフィールド名が変更された際、自動的に更新されます。

ストラクチャウインドウでテーブルまたはフィールド名を変更した場合、または、

4D コマンド「**SET TABLE TITLES**」または「**SET FIELD TITLES**」(「ユーザインタフェースコマンド」テーマ内) が呼び出された時

この機能は、フォーム内のテーブルやフィールド名をダイナミックに変更したい場合に特に役立ちます。

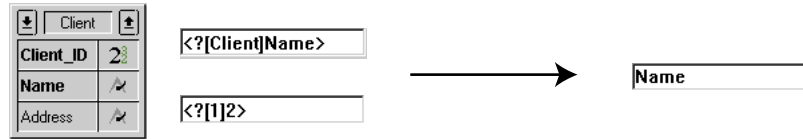
ダイナミックテーブル名またはダイナミックフィールド名をフォーム内に挿入するには、

- 1 「テキストエリア作成」ツールを使用してテキストを作成し、下記の命令を記述する。
 テーブル名の挿入：<?[テーブル名]>または<?[2]>（テーブルの作成順の番号で、2番目に作成されたテーブルを意味する）
 フィールド名の挿入：<?[テーブル名]フィールド名>または<?[2]3>（テーブルとフィールドの作成順の番号）または現在のテーブルのフィールドの場合は<?3>（フィールドの作成順の番号）

テーブルとフィールドの番号はこれらの作成順序に対応していることに注意してください。したがって、これらの記述を変更せずにテーブルとフィールドを追加することや名前を変更することができます。

2 テキストエリアの外側でクリックする。

ストラクチャウインドウで定義されると、現在のフィールドまたはテーブルの名前が現れます。



注：作成したテキストエリアの内容をフォームエディタ内で確認するには、そのエリア上をクリックするか、オブジェクトメニューからフォーマット表示または名前表示を選択することによって見るすることができます。

ユーザおよびカスタムモードで実行中の場合、テーブルまたはフィールドの名前 **SET TABLE TITLE** および **SET FIELD TITLES** コマンドを使って変更することができます。この場合、テーブルおよびフィールド名の参照は、これらのコマンドで設定された値を表示します。

ダイナミックフィールド名の使用は、フォームウィザードのオプションページでも指定できます。

デフォルトでは、ここは
チェックされていません。



Tips 内のダイナミックリファレンス

バージョン6.5では、変数やフィールドのTips内に現在のフィールドの値を挿入することができます。例えば、“ここに<[Family]Name>さんの年令を入力してください”というテキストを入れた場合、ユーザモードでは、[Family]Nameフィールドの現在の値が指定した位置に挿入されます。

リストのドラッグ&ドロップ

階層リストや階層ポップアップメニューをリストエディタやエクスプローラからドラッグ&ドロップすることにより、フォーム内に挿入することができます。

注：バージョン6.5では、リストに（従来の31文字ではなく）255文字まで含めることができます。

階層リストを挿入するには、リストエディタまたはエクスプローラの「リスト」ページからドラッグします。

階層ポップアップメニューを挿入するには、リストエディタまたはエクスプローラの「リスト」ページから、shiftキーを押しながらドラッグします。

注：同じ方法でピクチャライブラリからピクチャをドラッグ&ドロップすることによって、ボタンまたはピクチャメニューを作成することができます。詳細は、この章の「ピクチャライブラリ」を参照してください。

オブジェクトのバックグラウンドの管理

変数およびフィールドのバックグラウンドの管理が改善されました。

以前のバージョンでは、背景色として「背景色自動選択」を選択すると、バックグラウンドの透過効果を得ることができました。しかし、この色は実際にはフォームの背景色だったのです。

バージョン6.5では、変数およびフィールドに本当のバックグラウンド透過効果を適用できます。これを行うには、「パター無し」(Nの文字付のアイコン)を選択しなければなりません。この場合、変数およびフィールドのバックグラウンドは透過効果を得て表示されます。

オブジェクトプロパティ内で背景色として「背景色自動選択」を選択すると、データ入力の間はシステムによって定義された色が使用されます。

注：前景色と背景色は、ポップアップリスト、ドロップダウンリストおよびコンボボックスにも割り当てることができます。

カラーラベル

[?]ClientName>

バージョン6.5では、ボタン、チェックボックス、ドロップダウンリスト、スクロールエリア、サーモメータなどのラベルを持つすべてのオブジェクトで、ラベルの色を変えることができます。

オブジェクトのラベルの色を変えるには、次の方法のいずれかで色を適用します。

オブジェクトメニューまたはフォームエディタのコンテキストメニューで「カラー」を選択し、「描画色」の色を選択します。

オブジェクトプロパティパレットでは、カラーページで「描画色」の色を選択します。

プロパティリストでは、フォントカラー（「テキスト」テーマ）行で色を選択します。

プラットフォームインタフェース

プラットフォームインタフェースプロパティの管理が変更されました。

プラットフォームインタフェースリスト内（データベースプロパティ、フォームプロパティ、他）の「Copland」という用語が「プラチナ調」(Macintosh 8ルック)に置き換えられました。Macintoshでは、ダイナミックテキストオブジェクト（フィールドおよび変数）がフォーカスを持つと、エリアの周囲に陰付きフレームが表示されます。

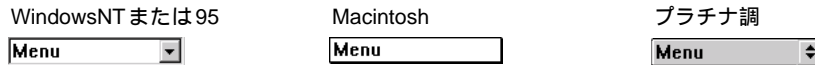
透過効果のある
バックグラウンド

会社名 Company name

階層リストの外観も、データベース、フォームまたはオブジェクトに定義されたインタフェースが適応されます。また、**SET PLATFORM INTERFACE** コマンドで定義されたパラメータも、**SET LIST PROPERTIES** コマンドが呼び出されていない限り有効です。

バージョン6.5では、ポップアップメニュー、ドロップダウンリストおよびドロップダウンメニューが、ポップアップ/ドロップダウンリストに統合されました。これら3つの名前はプラットフォームによってその外観が異なっているだけで、同じオブジェクトを意味しています。この新しいオブジェクトはプラットフォームのインタフェースプロパティ（データベース、フォームまたはオブジェクト用）により、よりシンプルに最適化されて使用されます。

フォーム内にポップアップ/ドロップダウンリストを挿入すると、その外観はプラットフォームによって異なります。



オブジェクトのプラットフォームインタフェースが「自動選択」の場合、その外観はフォームまたはデータベース用に定義されたプラットフォームインタフェースに応じて調整されます。

バージョン6.0.xのデータベースをバージョン6.5で開く場合に、以前の3つのプラットフォームインタフェースタイプのオブジェクトは外観が変更されないように調整され、互換性が保たれます。

フォーカス表示/入力可属性の変更

入力可能なオブジェクトは、フォーカス表示属性を使用することによって選択できなくなります。入力可能なフィールドおよびフォーカス表示が設定されていない変数には入力できません。

ピクチャボタン：2つの新しいモード

ピクチャボタンには「n Tick毎にスイッチ」と「ロールオーバーでスイッチ」の2つの新しいモードができました。

n Tick毎にスイッチ：このモードは、ピクチャボタンの内容を指定された速度で繰り返し変更させることができます。

オブジェクトプロパティパレットで、ピクチャボタンの構文で5番目のパラメータを渡すことで、このモードを指定します。このパラメータは、「n Tick毎にスイッチ」モードの指定を行うと共に、各ピクチャの切り替え時間（Tick）を設定します。

例えば、「2;3;?16807;0;10」を渡すと、ボタンは10Tickごとにピクチャの変更を繰り返します。

プロパティリストでは、「n Tick 毎にスイッチ」行（ディスプレイ & プリントテーマ）に値を入力します。

このモードでは、トランスポートピクチャオプション（モード64）を除いて他のすべてのオプションが無視されます。

ロールオーバーでスイッチ：このモードでは、マウスカーソルがピクチャボタンの上に移動すると、クリックしなくてもピクチャボタンの内容が変更され、マウスカーソルがピクチャボタンから離れると初期値のピクチャに変わります。

この動作は、ウェブナビゲータやマルチメディアアプリケーションでは頻繁に使用されています。最初表示されるピクチャは、「最後のフレームを無効として使う」オプション（モード128）も同時に指定されていなければ、フレームセットの最後のピクチャを使用します。この場合、ロールオーバーに使用されるのは最後から2番目のピクチャです。

オブジェクトプロパティパレット内で、ピクチャボタンの構文に16を渡すことにより、このモードを指定できます。

例えば、「action」、「roll over」、「disabled」モードを組み合わせたピクチャボタンを定義したい場合、「4;1;?15000;176」のように記述します。1行4カラムのフレームセットで、各フレームはaction/released、action/clicked、roll over、およびdisabledの状態に対応できます。

プロパティリストでは、「ロールオーバーでスイッチ」オプション（ディスプレイ & プリントテーマ）を選択します。

出力フォームのヘッダ

DISPLAY SELECTION および **MODIFY SELECTION** コマンドを使用して表示される、出力（リスト）フォームのヘッダ内に、アクティブオブジェクトを置いて使用することができます。

注：この機能は、ユーザモードで表示されるリストやサブフォームには適用されません。

アクティブでクリック可能な入力不可のオブジェクトのみが使用できます。

ボタン、デフォルトボタン、3D ボタン、ハイライトボタン、透明ボタン、ピクチャボタン

ポップアップ/ドロップダウンリスト、階層ポップアップメニュー、ピクチャポップアップメニュー

スクロールエリア、階層リスト

ラジオボタン、3D ラジオボタン、ピクチャラジオボタン

チェックボックス、3D チェックボックス

サーモメータ、ルーラ、ダイヤル

タブコントロール

注：コンボボックスは入力可能のため使用できません。

入力、キャンセル、スプリッタなどの標準の自動動作はボタンに関連付けられています。これらすべてのオブジェクトは、下記のフォームイベントを受け取ります。

- On Load (1)
- On Clicked (4)
- On Header (5)
- On Printing Footer (7)
- On Double Clicked (13)
- On Drop (16)
- On Drag Over (21)
- On Unload (24)

注：バージョン6.5には、新しいフォームイベントがあります。詳細は、第5章「ランゲージ」の「フォームイベント」を参照してください。

オンラインヘルプ

バージョン6.5は、カスタムオンラインヘルプファイルをそれぞれの4Dデータベースに関連付けることができます。更に、このヘルプファイルの各セクションをデータベースのフォームのそれぞれに関連付けできますので、オンラインヘルプの提供が可能になります。

カスタムオンラインヘルプファイルの作成

バージョン6.5のカスタムオンラインヘルプシステムは次のように働きます。

ヘルプファイルの名前は、データベースのストラクチャファイル名と同一でなければなりません。また、いかなるプラットフォームにおいても、ヘルプファイル（下記参照）の拡張子は「.HLP」または「.HTM」でなければなりません。

ヘルプファイルは、データベースのフォルダ内または「Win4DX」または「Mac4DX」フォルダ内に置かれなければなりません。

4D Serverでヘルプファイルをすべてのクライアントで使用したい場合、ファイルを「Win4DX」または「Mac4DX」フォルダ内に置いてください。

ヘルプファイルは下記のタイプのうちのひとつを使用することができます。

Windows 標準のマイクロソフトヘルプ¹フォーマットである「.HLP」ファイルは、Macintosh用のマイクロソフトヘルプがインストールされていればMacintosh²でも使用できます。

1.「.HLP」ファイルを作成する方法については、Microsoftのドキュメントを参照してください。

2. ファイルのタイプを「HELP」に、クリエータを「MSHE」に変更しなければなりません。ファイルをきちんと変換できるかどうかは、オンラインヘルプコンパイラのバージョンと、使用されているマイクロソフトヘルプのバージョンに依存します。それ以外の場合は、オンラインヘルプを使用することはできません。

HTML フォーマットである「.HTM」は、すべてのプラットフォームに同一のオンラインヘルプを作成でき、Web ブラウザ内にも表示することができます。

アップルガイドファイル（Macintosh の場合のみ）は、coachmarks（アプリケーションウインドウ内でオブジェクトをコーチすることのできる）を使用します。アップルガイドがインストールされていない場合、4D はマイクロソフトヘルプフォーマットでファイルを開こうとします。

注：Plug-in もヘルプファイルを持つことができます。これは、シングルユーザおよびクライアントサーバの両方のアプリケーション用に、「Win4DX」または「Mac4DX」フォルダ（必要であれば両フォルダ）内に置かれていなければなりません。Plug-in のヘルプファイルは、Plug-in 名に「.HLP」または「.HTM」拡張子の付いた名前であればなりません。

オンラインヘルプの作成

4D データベース用のオンラインヘルプの作成は、セクション番号をフォームのそれぞれに関連付けることから始まります。それにより、フォームから対応するヘルプページを呼び出すことができるようになります。

下記条件を満たすと、オンラインヘルプが使用できるようになります。

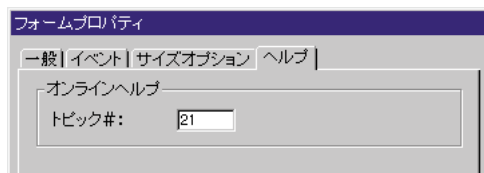
Windows では、オンラインヘルプが HLP フォーマットである場合。

Macintosh では、オンラインヘルプが Apple Guide フォーマットである場合。

すべてのプラットフォームでは、オンラインヘルプが HTML フォーマットの場合、また Web ブラウザが Internet Config 1.2 と互換性のある場合。

セクション番号を 4D フォームに関連付けるには、

- 1 フォームエディタ内で、フォームメニューから「フォームプロパティ」を選択する。
フォームプロパティが現れます。
- 2 ヘルプタブをクリックする。
このプロパティは、プロパティリストでもアクセス可能です。
- 3 フォームに関連付けたいセクション番号を入力する。



4 「OK」ボタンをクリックする。

5 ヘルプファイルには、このフォームに関わりのあるセクションに同じ番号を宣言する。
この操作は、使用しているフォーマットにより変わります。

HLPファイルについては、マイクロソフトオンラインヘルプコンパイラのドキュメント(Help Compiler)を参照してください。

アップルガイドについては、Appleより提供されているドキュメントを参照してください。

HTMLファイルでは、各セクションを宣言して、それに番号を割当てする必要があります。

セクションは、次のタイプのマーカーを使用して宣言します。

```
<A NAME="Section">
```

例えば、

```
<A NAME="21">
```

セクションのURLは次のような形式です。

```
<A HREF="#Section">...</A>
```

フォームに設定されたセクション番号が0の場合、4Dはヘルプファイルの最初のページを表示します。

4D データベースからオンラインヘルプの呼び出し

4Dは2つの方法でデータベースのカスタムオンラインヘルプを呼び出すことができます。

ヘルプメニューから「データベース名ヘルプ」を選択する方法（Macintoshのバージョン7では、このメニューは？マークとして表します）。この場合、ヘルプファイルの最初のページが表示されます。

フォームが表示されている時に、F1キー（MacintoshおよびWindowsの場合）またはHelpキー（Macintoshの場合のみ）を押す方法。この場合、ヘルプフィールド番号がフォームに関連付けられていれば、対応するページが表示されます（オンラインヘルプ）。そうでない場合はヘルプファイルの最初のページが表示されます。

注：4th Dimensionは独自のオンラインヘルプシステム（4Dエンジンと4Dを組み込んでコンパイルした場合を除き、ヘルプメニューからアクセス可能）を持っています。Windowsではオンラインヘルプがメソッドエディタ内で使用できます。コード内のコマンドを選択してF1キーを押すと、コマンドの内容が現れます。

ランタイムエクスプローラ

この新しいウインドウを利用することで、データベース内の構成要素を監視し、それらが正しく利用されているかどうかを知ることができます。特に、データベースの開発や分析時に役立ちます。

ウインドウの表示

ランタイムエクスプローラのウインドウは、デザインモード、ユーザモードおよびカスタムモード（コンパイルされているかどうかに関わりなく）の4th Dimensionの全モードでアクセス可能です。

注：カスタムモードでは、デザイナーと管理者のみがランタイムエクスプローラウインドウにアクセスできます。

ランタイムエクスプローラは、「従来の」ウインドウ（デザインモードのみ）またはフローティングパレット（すべてのモード）の2つのタイプのウインドウで表示できます。フローティングパレットは常に前面に表示されます。

ランタイムエクスプローラをウインドウ内に（デザインモードで）表示するには、

- 1 ツールメニューからランタイムエクスプローラを選択する。

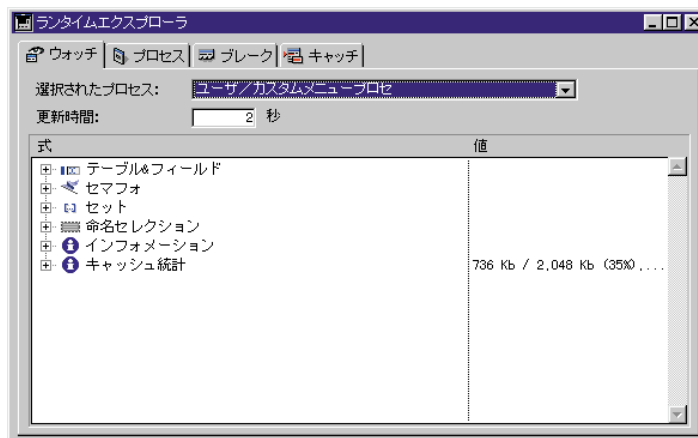
フローティングパレット内に（デザイン、ユーザ、カスタムの各モードで）ランタイムエクスプローラを表示するには、

- 1 Windows では、「Ctrl+Shift+F9」キーを押す。
Macintosh では、「command+shift+F9」キーを押す。

または、

shiftキーを押しながら、ツールメニューからランタイムエクスプローラを選択する（デザインモードのみ）。

ランタイムエクスプローラウインドウには、ウォッチ、プロセス、ブレイク、キャッチのタブ上をクリックすることによってアクセスできる4つのページがあります。



ウォッチページ

ウォッチページは、コードの実行に関する情報を表示します。

注：このページに表示される情報は、4D デバッガのウォッチ枠内の情報と同一のもので
す。

選択されたプロセス：このドロップダウンリストには、データベース内で実行されて
いるすべてのプロセスが含まれており、見たいプロセスを選択することができます。

更新時間：ページ内の情報がどの程度頻繁に更新されるかを、秒単位で指定するこ
とができます。

「式」カラムはオブジェクトの名前と式を表示します。「値」カラムはオブジェクトと式
に対応する現在値を表示します。カラム間の境界線をドラッグすることで、カラムのサ
イズを変更することができます。

変更可能なオブジェクトは、右カラム内の値をクリックすることによって値を変更する
ことができます。

マルチレベルリストは、複数のテーマで構成されます。テーマは下記の通りです。

プロセス：実行中のプロセスのリストや現状を見ることができます。

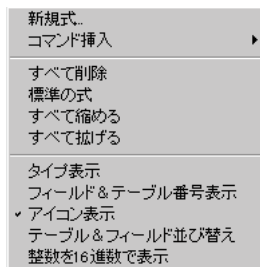
変数：インタープロセス変数のリストおよび選択されたプロセスのプロセス変数のリ
ストを見ることができます。

テーブル&フィールド、セマフォ、セット、命名セクション、インフォメーショ
ン：これらのテーマ内で提供される情報は、4D のデバッガで提供される情報と同一の
ものです。詳細は、『4th Dimension ランゲージリファレンス』の「デバッグ」の章を参
照してください。「テーブル&フィールド」テーマには、本書の「デバッガ」で説明し
ている追加情報が含まれています。

キャッシュ統計：4D のキャッシュについての情報が得られます。この新しいリストは、
バージョン 6.5 のデバッガでも得られます。

式またはテーマを削除するには、対応する行を選択して delete キーを押します。

また、「新規式...」、「コマンド挿入」、「すべて削除」、「標準の式」、「すべて縮める」、「す
べて広げる」などを実行させることもできます。



それらを実行させるには、コンテキストメニュー（Windowsではマウスの右ボタン、MacintoshではControlキーを押しながらクリック）内の対応するコマンドを選択します。

更に、コンテキストメニューの下部にはいくつかの表示オプションが表示されています。詳細は、『4th Dimension ランゲージリファレンス』の「デバッグ」の章を参照してください。

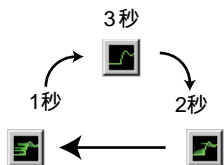
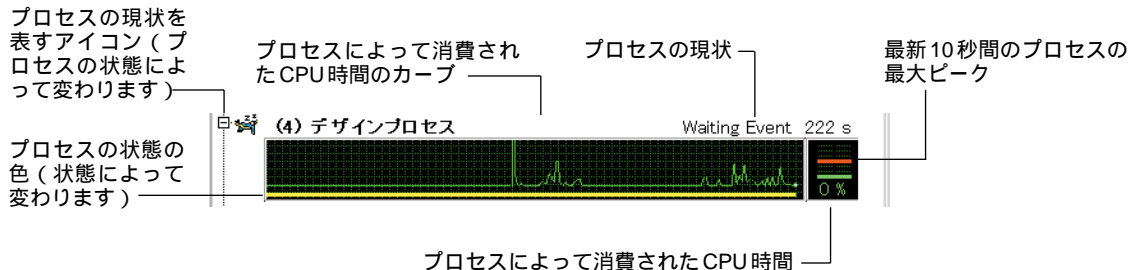
プロセスページ

プロセスページでは、各プロセスによって消費された時間と各プロセスの状態を視覚的に見ることができます。

注：このページは、以前のバージョンの「プロセスリスト」に変わるもので、ユーザーモード、カスタムモード並びにコンパイルされたデータベース内で呼び出すことができる利点があります。さらに、このページは新しい機能も備わっています。

プロセス名の左の拡張アイコンをクリックすることによって、各プロセスのグラフィック表示/非表示を指定できます。また、ウインドウのツールバー内の「表示」または「隠す」ボタンをクリックすることによって、全アイコンの表示/非表示を指定することができます。

表示された情報の内容を示します。



ウインドウの右上にあるアイコンを連続的にクリックすることで、データが更新される頻度を1秒、2秒または3秒のいずれかに変えることができます。更新頻度が多ければ、より多くのCPU時間がランタイムエクスプローラプロセスによって消費されます。グラフィックで画面上に表示するプロセスの数は、プロセスによって消費されるCPU時間にも影響を与えます。

注：グラフィック表示が閉じられている時には、CPU時間を消費していません。

グラフィックエリア内でクリックするとクリックした場所に縦の線が現れ、その時点でのプロセスの状態が表示されます。マウスボタンを押して横に移動させると、プロセス状態の進展を見ることができます。

プロセス管理コマンドには、ウインドウ内のツールバーの下記のボタンを使用してアクセスすることができます。

実行 、一時停止 、アボート 、トレース 、隠す 、表示 、前面に 

ブレイクおよびキャッチページ

ブレイクページは、データベース内に置いたブレイクポイントを表示します。

キャッチページは、コマンド（または式）の検知を定義されたブレイクポイントを表示します。

注：これらのページは、以前のバージョン内のブレイクリストの「ブレイクポイント」と「4D コマンドの検知」に置き換わるもので、ユーザモード、カスタムモードおよびコンパイルされたデータベース内で呼び出すことができるという利点があります。

これらのページの一般的な機能は、4D のブレイクリストと同一のものです。詳細は、『4th Dimension ランゲージリファレンス』の「デバッグ」の章を参照してください。ブレイクリストメニュー内のコマンドは、ウインドウのツールバーのボタンに置き換えられました。

注：バージョン 6.5 では、4D メソッドエディタ内でブレイクポイントを直接定義することができます（詳細は、この章の「ブレイクポイントの挿入」を参照してください）。

オブジェクト検索エディタ

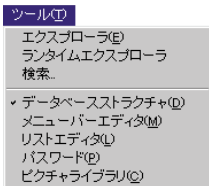
オブジェクト検索エディタはバージョン 6.5 の新しい機能であり、これは 4D Insider にあった機能です。このエディタは、データベースストラクチャ全体から指定した文字列を探することができます。

データベース内の検索

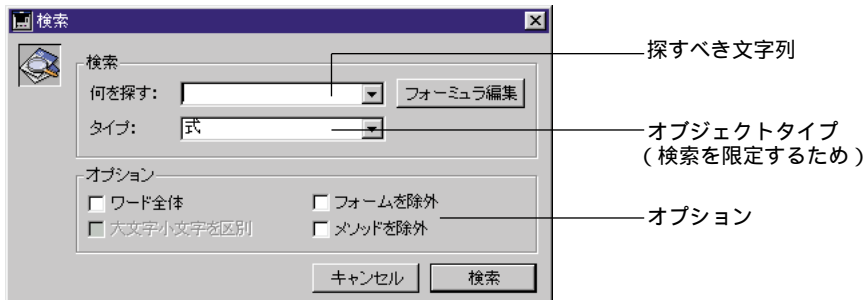
バージョン 6.5 のオブジェクト検索エディタは、デザインモードで 사용할 ことができます。

文字列を見つけるには、

- 1 デザインモードで、ツールメニューから「検索...」を選択する。



オブジェクト検索エディタが現れます。



2 「何を探す」エリアに、探すべき文字列を入力する。

注：@ は文字として見なされ、データベースストラクチャ内で文字列を探す時にワイルドカードとして使用することはできません。

3 必要あれば、検索を限定したいオブジェクトタイプを定義する。

「検索」を特定のオブジェクトタイプに限定すると、素早い検索が可能になります。検索するオブジェクトタイプを限定しなくなれば、タイプメニュー内のすべてを選択します。

タイプメニュー内で式を選択すると、フォーミュラ編集ボタンが使用可能になります。これは、4Dの「フォーミュラ」内を直接検索することができるようにするものです。

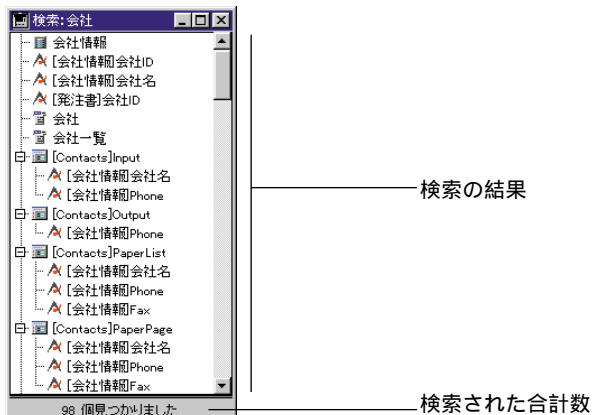
オブジェクトタイプは、「オブジェクトタイプと検索のスコープ」に記述しています。

4 必要があれば、検索のオプションを定義する。

これらのオプションについては、「検索のオプション」に記述しています。

5 「検索」ボタンをクリックするか、またはenterキーを押す。

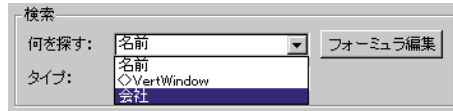
検索が終了すると、検索された内容が新しいウインドウに階層リストとして表示されます。



オブジェクトを適切なエディタで見るには、このウインドウ内の行をダブルクリックします。

複数の検索を行なうと、各検索の結果ウインドウを開きます。

検索を実行すると、「何を探す」エリアに入力された値は次回からはコンボボックスから選択できるようになります（ただし、同一セッション内でのみ）。



同じ検索を何回も速やかに始めることもできます。

オブジェクトタイプと検索のスコープ

オブジェクト検索エディタで、検索するオブジェクトタイプを定義することができます。

式：例えば、total:=Sum([Accounts]Total)

変数：例えば、\$vpPicture1

テーブルまたはフィールド名：例えば、[Clients]Name

フォームオブジェクト名：例えば、Background

コメント：例えば、Modified on

他のすべてのオブジェクトタイプのグローバル検索を実行することもできます。

デフォルトでは、データベースストラクチャ全体を通して文字列が検索されますが、検索からメソッドやフォームを除外することもできます。

指定されたオブジェクトタイプによっては、下記のエレメントの間で行われます。

フォーム（除外可能）

メソッド（除外可能）

メニューおよびメニューコマンド

リスト

テーブルおよびフィールド（同様に、サブテーブルおよびサブフィールド）

コメント

下記テーブルは、異なるオブジェクトタイプに応じて、検索が可能な構成エレメントを示します。

| | | 検査を行なう場所 | | | | | |
|-----------|---------------|--------------|--------------|---------------|-----|---------------------------|------|
| | | フォームおよびフォーム名 | メソッドおよびメソッド名 | メニュー/メニューコマンド | リスト | テーブルおよびフィールド(ストラクチャウインドウ) | コメント |
| オブジェクトタイプ | 式 | | | | | | |
| | 変数 | | | | | | |
| | テーブルおよびフィールド名 | | | | | | |
| | フォームオブジェクト名 | | | | | | |
| | コメント | | | | | | |
| | すべて | | | | | | |

検索のオプション

オプションをチェックボックスで選択することができますが、検索のタイプによっては選択できないものもあります。

ワード全体

このオプションが選択されると、完全一致で検索します。例えば、「client」を検索する場合、4Dは「clients」または「myclient」を対象外とします。

デフォルトでは選択されていないため、「var」の検索を行うと「Myvar」、「variation」なども探します。

大文字小文字を区別

このオプションは、ワード全体のオプションが選択されている場合にのみ有効です。このオプションが選択されていると、大文字小文字を区別するため、「MyVar」を検索する時には、「myVar」は対象外となります。

フォームを除外

このオプションが選択されると、フォーム内およびフォーム名を除いてデータベース全体の検索が行われます。

メソッドを除外

このオプションが選択されると、メソッド内およびメソッド名を除いてデータベース全体の検索が行われます。

コメント

バージョン6.5は、データベースのオブジェクトにコメントを付けることができます。コメントの使用は複数のプログラマによって開発されているデータベースに特に適したものであり、データベースコードのメンテナンスに有効です。

4Dのコメントは、テキストとして表示（異なるフォントスタイルや色などが指定可能）され、そのテキストはデザインモードでいつでも変更することや見ることができます。これは、オブジェクトの内容とデータベース内でどのように機能するかを理解するために必要な情報が含まれています。作成したコメントはデータベースストラクチャに保存されます。

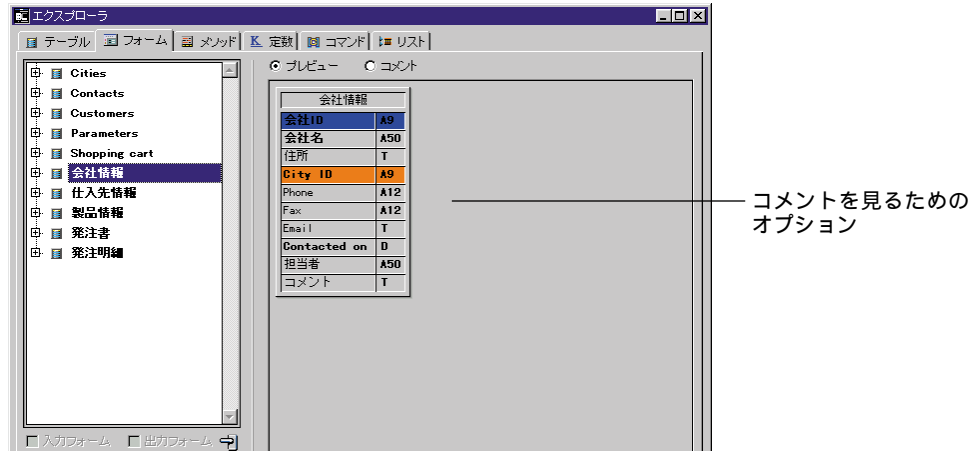
さらに、バージョン6.5では自動的にコメントを作成することもできます。オブジェクトが作成または変更されると、4Dが自動的にコメントを入力するということです。

注：バージョン6.5で作成されたコメントは、4D Insiderのコメントと互換性があります。

メソッド（データベースメソッド、プロジェクトメソッド、トリガおよびフォームメソッド）、フォーム、テーブル、サブテーブルおよびフィールドにもコメントを付けることができます。

オブジェクトにコメントを付ける

エクスプローラからのコメントの作成、変更、参照が可能です。エクスプローラのプレビューエリアには、プレビューおよびコメントの2つのラジオボタンがあります。



プレビューオプション（デフォルトで選択）は、以前のバージョンと同じようにエクスプローラの標準機能に対応します。

コメントオプションが選択されると、プレビューエリアは、コメントエリアに置き換わります。



コメントエリアには垂直スクロールバーがあり、テキストをスクロールできます。

4D Server : エリア内の左下にあるロックアイコンは、そのコメントが他のユーザによって編集かどうかを示すものです。その場合には、ペンシルにはスラッシュがかり、コメントは参照のみが可能となります。

コメントの作成、変更、参照するには、

- 1 エクスプローラの左側の部分で、オブジェクト（メソッド、テーブルまたはフィールド）を選択する。
- 2 「コメント」ラジオボタンをクリックする。
選択されたオブジェクト用のコメントが現れます。
- 3 コメントエリアにテキストを入力する。
コメントは入力エリア以外をクリックすると、すぐに登録されます。各オブジェクトには32KB（32000文字）までのコメントを入力できます。

標準のテキスト編集コマンド（コピー、ペースト、すべてを選択など）を使用できます。また、他のテキストエリアと同様にキーボードショートカットを使って、コメントエリア内のテキストを操作することができます。

コメント用フォント属性の変更

コメントのフォントスタイルを太字やイタリックで強調する、別のフォントを選ぶ、色を変更するといったことができます。

コメントのフォントスタイルを変更するには、

- 1 コメントエリアで変更したい部分を選択する。
- 2 Windowsではマウスの右ボタンでエリア内をクリックする。
Macintoshではエリア内をControlキーを押しながらクリックする。

階層ポップアップメニューが現れます。



- 3 テキストに適用したいフォント属性を選択する。

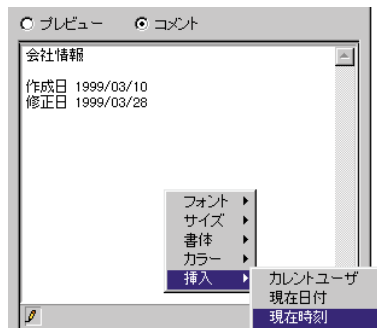
日付、時間またはユーザ名の挿入

バージョン6.5は、日付、時間またはユーザ名（4Dのパスワードエディタで定義されたもの）などのような標準的情報をコメント内に挿入するショートカットを提供します。

注：このショートカットは、自動的なコメントの作成に適したものです（下記参照）。手動でのコメント入力を促すためにも使用できるでしょう。

日付、時間またはユーザ名をコメント内に挿入するには、

- 1 コメントエリア内で情報を挿入したい位置をクリックする。
- 2 Windowsではマウスの右ボタンでエリア内をクリックする。
Macintoshではエリア内をControlキーを押しながらクリックする。
階層ポップアップメニューが現れます。



- 3 挿入したい情報を選択する。
選択された情報の現在の値がコメントエリアに挿入されます。
データベースにパスワードが設定されていない場合、ユーザ名は現れません。

注：このショートカットはデータ入力を助けます。しかし、変数を挿入することができないため、挿入された情報は自動更新されません。自動コメントを挿入するには、次項を参照してください。

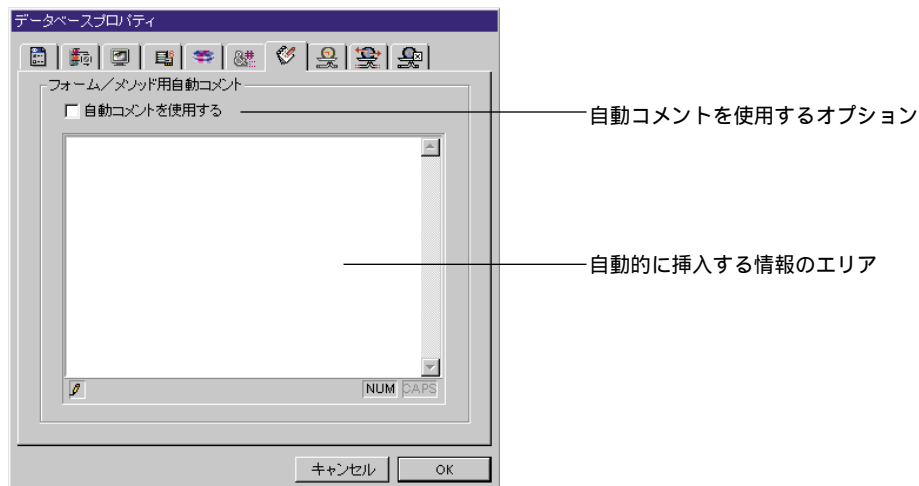
自動コメントの挿入

自動コメントシステムを起動させることができます。ただし、メソッドおよびフォームのみに使用可能です。

このシステムが起動すると、データベース内で作成または変更されたメソッド、あるいはフォームに対し、自動的にコメントが作成されます。

自動コメントシステムを起動させるには、

- 1 「データベースプロパティ」ダイアログ内で、「コメント」タブをクリックする。
次のようなページが現れます。



- 2 「自動コメントを使用する」オプションを選択する。
- 3 下の入力エリアに、自動的に挿入したい情報を入力する。
コメント内に挿入される前に更新される変数を使用することができます。

日付は、#D

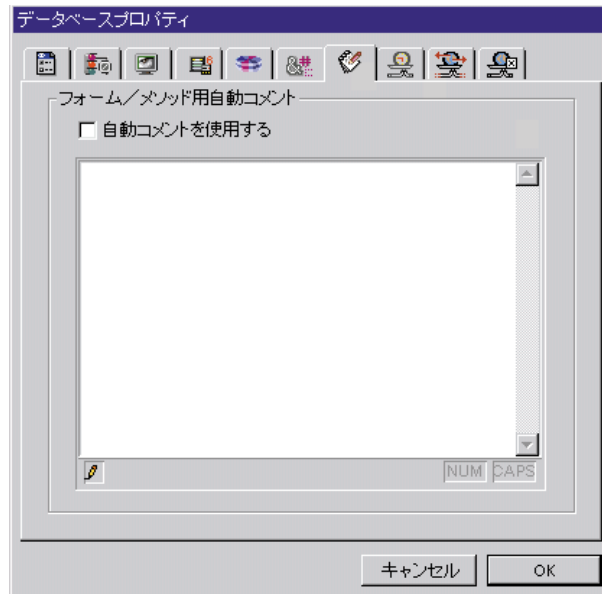
時間は、#H

現在のユーザは、#N

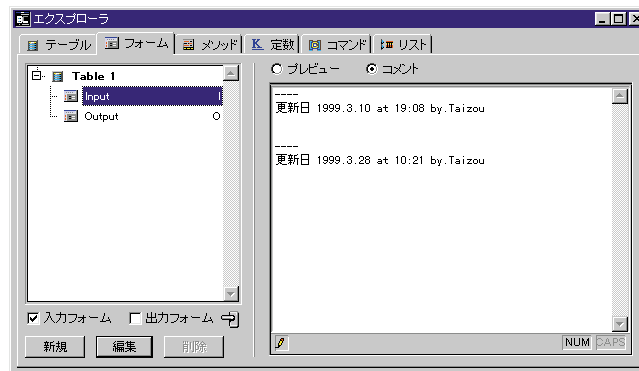
注：データベースにパスワードが設定されていない場合、#Nは空の文字列を返します。

ポップアップメニューを使ってもこの情報を挿入することができます（詳細は、この章の「日付、時間またはユーザ名の挿入」のステップ2を参照してください）。

例えば、下記の値を入力した場合には、



作成または変更されたすべてのメソッドおよびフォームのコメントエリアが更新されます。



4D Server : 自動コメントは、データベースプロパティにアクセス権を持っているどのクライアントでも変更することができます。これらのパラメータをサーバ側からも変更することができます。

自動コメントページに実施された変更は、各クライアントでオブジェクトが変更されると、即座に取り入れられます。

メソッドの編集

バージョン6.5にはメソッド記述を手助けする新しい機能が付きました。ランタイムエクスプローラ内から、コマンドシンタックスをメソッドエディタにプレ入力することができます。また、テキスト編集を操作する新しいショートカットもあります。使用されるフォントを定義することや、メソッドエディタ内に直接ブレークポイントを設定することもできるようになりました。

シンタックスヘルプ

新しいシンタックスヘルプの機能が3つあります。

4D コマンドのシンタックスを見ることができます。

シンタックスと共に4D コマンドを挿入することができます。

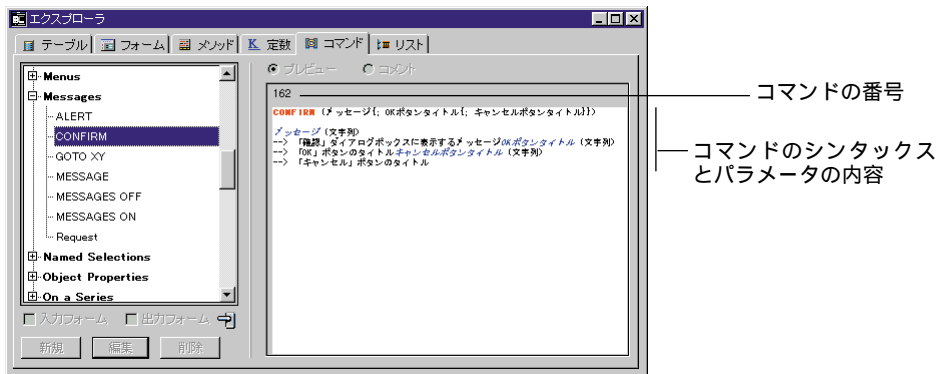
メソッドエディタのウインドウ内で直接シンタックスエラーを知らせます。

シンタックスを見る

4D コマンドのシンタックスを、エクスプローラ内またはメソッドエディタ内で見ることができます。

エクスプローラ内でシンタックスを見る

エクスプローラ内で4D コマンドのシンタックスを見るには、コマンドタグで表示されるコマンド名をクリックします。プレビューエリアに、コマンドのシンタックス、パラメータの簡単な内容およびそのコマンドが記されているランゲージマニュアルのページを表示します。



メソッドエディタ内でシンタックスを見る

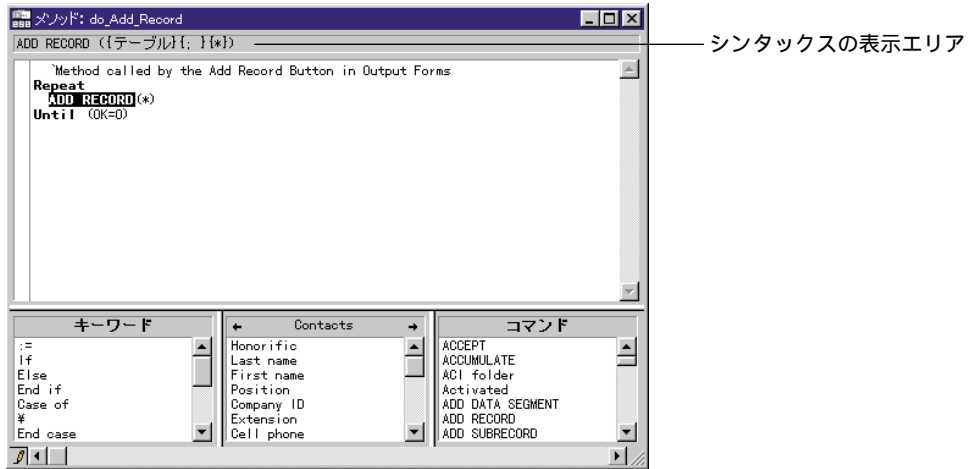
メソッドエディタ内でコマンドのシンタックスを見るためには、下記の2つの方法があります。

コマンドリスト内のルーチンを選択する

編集エリア内でコマンドを入力または選択し、Enterキーを押す。

注：メソッドエディタ内でコマンドを選択し、F1キーを押すと、選択されたコマンド用に4DのWindowsオンラインヘルプを表示できることを覚えておいてください。

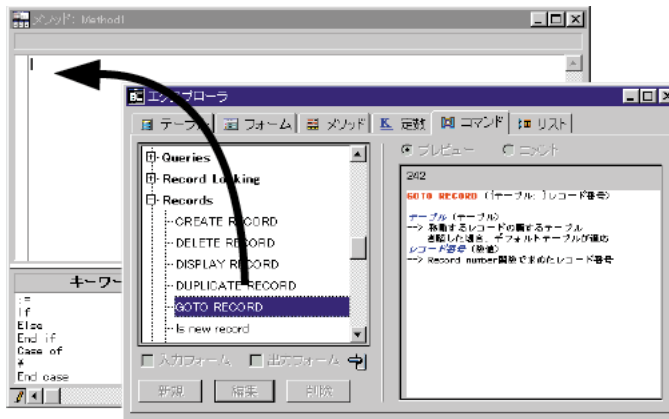
メソッドエディタウインドウ上部の新しいエリア内にシンタックスが表示されます。



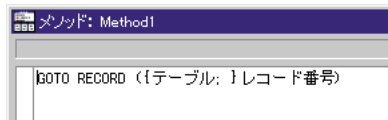
コマンドのシンタックスをプレ入力する

コマンドシンタックスは、メソッドエディタ内にすべてのパラメータが付いた形式で挿入することができます。挿入されたシンタックスは、コードに適した内容に変更できる「インプットマスク」として、より多く表されます。この特長により、メソッドを書いている間にパラメータを忘れることがないでしょう。

コマンドを、そのシンタックスと共に挿入するには、エクスプローラからドラッグしてメソッドエディタ内にドロップする必要があります。



デフォルトでは、コマンドはそのシンタックスと共に挿入されます。



以前のバージョンのようにシンタックスをブレ入力しないで、コマンドをドラッグ&ドロップしたい場合には、WindowsではAltキーを、Macintoshではoptionキーを押しながらドラッグ&ドロップします。

注：コマンドのシンタックスを見てブレ入力するには、ディスク内に4D Helpファイルが存在している場合のみに限られます。これらの機能を活かすには、ヘルプファイルを移動したり、名前を変更していないことを確認してください。

- ・ Macintoshでは、4D Help65Jファイルはシステムフォルダの「初期設定：ACI」フォルダ内または4Dアプリケーションのフォルダ内に置かれていなければなりません。
- ・ Windowsでは、4D Help65J.RSRファイルが「Windows¥ACI」フォルダ内（WindowsはWindowsシステムファイルフォルダを表わします） または4Dアプリケーションのフォルダ内に置かれていなければなりません。

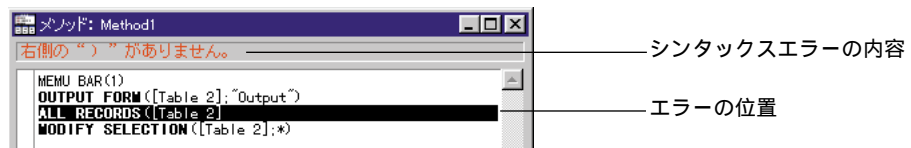
シンタックスエラーを見る

シンタックスエラーは、メソッドエディタのウインドウに直接示されるようになります。4Dは行（ステートメント）またはメソッド全体が受け入れられた時に、シンタックスを検査します。

メソッド全体の検査は、ウインドウを保存するか閉める時に自動的に行われます。「Ctrl+Enter」キー（Windowsの場合）「command+enter」キー（Macintoshの場合）を押すことによって、メソッドの検査を強制的に実施することができます。

検査の際に、4Dは各ステートメントのコマンドシンタックスやプログラミング構造（If...,End if など）をチェックします。

エラーが発見された場合には、メソッドウインドウ上部の情報エリアにメッセージが書き込まれ、エラーを含んでいるステートメントを選択します。



テキスト編集を操作するショートカット

メソッドエディタでテキスト編集を操作するのにキーボードのショートカットが使えます。これらのショートカットは、データ入力エリアを持つすべての4Dのダイアログでも使えます。

| Windows | MacOS | 動作 |
|----------------|-------------------|--|
| shift + | | 選択範囲を文字単位で右側へ移動 |
| shift + | | 選択範囲を文字単位で左側へ移動 |
| shift + | | 行単位で上から下への選択範囲の拡大 |
| shift + | | 行単位で下から上への選択範囲の拡大 |
| Ctrl + Shift + | command + shift + | 単語単位に右側からの選択範囲の拡大 |
| Ctrl + Shift + | command + shift + | 選択範囲を単語単位で左側へ移動 |
| Ctrl + | command + | 単語単位に左から右へ挿入ポイントを移動 |
| Ctrl + | command + | 単語単位に右から左へ挿入ポイントを移動 |
| | home | 行の最初に挿入ポイントを置く |
| | end | 行の最後に挿入ポイントを置く |
| Ctrl + Home | command + home | メソッドの最初に挿入ポイントを置く |
| Ctrl + End | command + end | メソッドの最後に挿入ポイントを置く |
| shift + home | | 行内でカーソルの左側にある全ての文字を選択する |
| shift + end | | 行内でカーソルの右側にある全ての文字を選択する |
| pageup | | メソッドの内容をページ単位で下から上へスクロールする（挿入ポイントは変更しない） |
| pagedown | | メソッドの内容をページ単位で上から下へスクロールする（挿入ポイントは変更しない） |

メソッドエディタ内でのフォントの定義

「データベースプロパティ」ダイアログ内で、メソッドエディタで使用されるフォントを定義することができます（以前はCustomizer Plusでのみ定義可能でした）。

メソッドエディタ内で使用されるフォントを定義するには、

- 1 「データベースプロパティ」ダイアログ内で「ユーザインタフェース」タブをクリックする。
- 2 メソッドエディタ内で使用したいフォントおよびフォントサイズ（ピクセルで）を定義する。
- 3 「OK」ボタンをクリックする。
変更はすぐに適用されます。

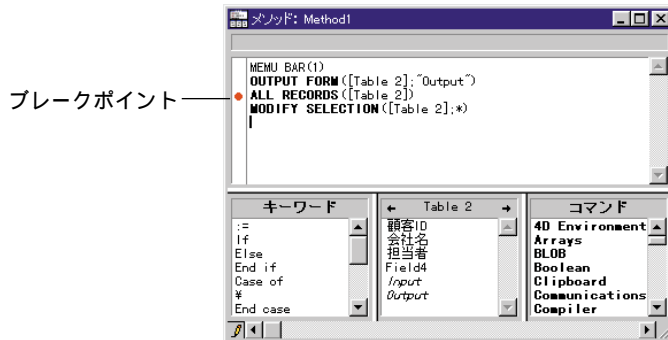
ブレイクポイントの挿入

メソッドエディタ内で直接ブレイクポイントの挿入、変更、削除をすることができます。もちろん、ブレイクポイントはメソッドエディタとランタイムエクスプローラの間で（デバッガも同様に）お互いに影響し合っています。実際、ブレイクポイントはこれらの3つのエディタ内で定義することができます。

注：ランタイムエクスプローラに関する詳細は、この章の「ランタイムエクスプローラ」を参照してください。

メソッドエディタ内にブレイクポイントを加えるには、

- 1 ブレイクポイントを挿入したい左カラム内でクリックする。
このカラムの上にマウスポインタが置かれると、点の形に変わります。クリックすると、赤い丸が左のカラムに現れます。



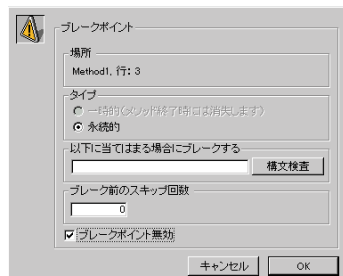
注：コードの行を挿入または削除しても、ブレイクポイントはメソッドエディタ内では常に同じ場所に留まります。

メソッドエディタ内でブレイクポイントを削除するには、

- 1 削除したいブレイクポイントをクリックする。
ブレイクポイントが消えます。

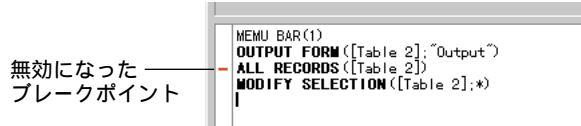
メソッドエディタ内でブレイクポイントを無効にするには、

- 1 Altキー(Windowsの場合)またはoptionキー (Macintoshの場合)を押しながら、ブレイクポイントをクリックする。
ブレイクポイントプロパティのダイアログが現れます。



注：このダイアログ、および一般的なブレークポイントに関する詳細は、『4th Dimension ランゲージリファレンス』の「デバッグ」の章を参照してください。

- 2 ブレークポイントを無効にするオプションをチェックし、「OK」ボタンをクリックする。



ブレークポイントは無効になりました。

新しいデータベースプロパティ

バージョン6.5のデータベースプロパティに、新しいオプションが追加されました。パスワード（「パスワード」に記載）およびWebサーバ（「Webサーバ」に記載）に関連したプロパティとは別に、次のようなプロパティを設定することもできます。

文字「@」の管理

テンポラリフォルダの位置の定義

スタートアップでのクライアントの自動登録（4Dサーバ）

注：後述されている新しいルーチン **SET DATABASE PARAMETER** を使用する場合は、一般のパラメータにもアクセスが可能です。

文字「@」の管理

検索や文字列比較で「@」が単語の間にある場合、4Dが「@」をどのように取り扱うかを定義することができます。これは、「ワイルドカード」（詳細は4『4th Dimension ユーザリファレンス』を参照）として、または単なる文字として取り扱うように定義することができます。

この指定は、Eメールアドレス（通常は、name@provider.xx）を保存するデータベースで特に役立ちます。

「@」文字をどのように取り扱うかを定義するには、

- 1 データベースプロパティで、2番目のタブ「データ制御&アクセス権」をクリックする。対応するページが現れます。デフォルトでは、クエリ/並び替えで「@」を文字とみなすオプションはチェックされていないため、以前のバージョンのように「@」は「ワイルドカード」として定義されます。

このオプションを選択した場合、「@」が文字列の中に見つめられると、通常の文字として取り扱われます。

このオプションは、検索、ソート、文字列の比較でテーブル内に保存されたデータや配列などのようなメモリ内のデータに影響を与えます。文字列タイプ（インデックスが付けられているもの）のフィールドと変数およびテキストに対しても関わります。

注：

- ・ 検索対象文字が「@」で始まるか、または「@」で終る場合には、「@」はワイルドカードとして取り扱われることに注意してください。「@」が単語の中にある場合（例えば、bill@cgi.com）にのみ、異なった方法で扱います。
- ・ このオプションは、ワイルドカード文字「@」をオブジェクトパラメータ内に受け入れる「オブジェクトプロパティ」テーマ内のコマンドの動作にも影響を与えます。『4th Dimension ユーザリファレンス』を参照してください。

2 オプションを指定し、「OK」ボタンをクリックする。

3 データベースを閉じて再度開く。

このステップは、オプションを有効にするために必要です。

データベースが再度開かれると、自動的にすべてのインデックスを再構築します。

テンポラリフォルダをどこに置くかを定義する

4D がデータベースを実行中に作成するテンポラリファイルの保存位置を指定することができます。4D はテンポラリファイルを主にトランザクションや命名セクションに使用します。

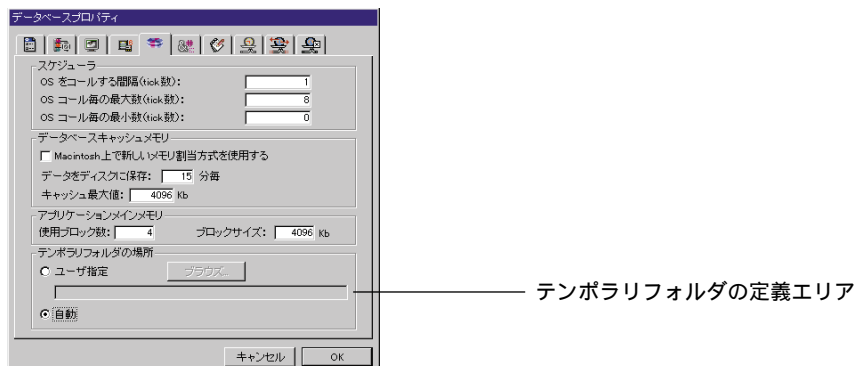
デフォルトは、以前のバージョンのように自動的に管理されます。

Windows では、4D はテンポラリファイルを C: ドライブに置きます。

Macintosh では、4D はテンポラリファイルを（ボリューム位置を考慮することなく）最も空き領域のあるボリュームに置きます。

テンポラリフォルダの位置を変更するには、

- 1 データベースプロパティで「システム設定」タブをクリックする。
次のページが現れます。



デフォルトでは、自動オプションが選択されます。

- 2 「ユーザ設定」ラジオボタンをクリックする。
「ブラウズ...」ボタンが有効になります。
- 3 「ブラウズ...」ボタンをクリックする。
標準の「ファイル作成」ダイアログが現れます。
- 4 テンポラリフォルダを置きたい位置を選択し、「OK」ボタンをクリックする。
トランザクションや命名セレクションのテンポラリファイルは指示された場所書き込まれます。アクセスパスは、4Dの初期設定ファイルに保存されます。

注：アクセスパスが正しくない場合は、自動モードが（データベースプロパティ内でオプションを変更することなしで）設定されます。アクセスパス（ボリュームまたはフォルダの名前）が変更されていないことを確認するのは、データベース開発者の責任です。

スタートアップ時にクライアントを登録する

この新しいオプションでは、クライアントが4D Serverに接続するとすぐに、クライアントの自動登録を可能にします。一旦「登録」されると、クライアントは他のクライアントまたはサーバが要求することをすべて実行することができます（後述のルーチン **REGISTER CLIENT** の内容も参照してください）。

このオプションは、クライアントまたはサーバのプロパティ内で定義することができ、データベースに接続する各クライアントに適用されます（このオプションはストラクチャファイルに保存されます）。

注：バージョン6.5で作成されたデータベースには、このオプションがデフォルトで選択されます。以前のバージョンで作成されたデータベースには、デフォルトで選択されません。

クライアントを自動的に登録するには、

- 1 データベースプロパティで「接続設定」タブをクリックする。
- 2 「開始時にクライアント登録」オプションを選択して「OK」ボタンをクリックする。
これで、データベースに接続するすべてのクライアントは自動的に登録されます。しかし、既に接続しているクライアントにこのオプションを適応するには、一旦クライアントを終了し再接続させる必要があります。

このオプションは、主にユーザモードでの「メソッド実行」ダイアログ内の **On 4D Client** 関数で使用されます（詳細は、第3章「ユーザモード」の「メソッドの実行」を参照してください）。クライアント間でタスクを分配する洗練されたシステムを作成するには、この目的のために作成されたコマンドを使用することが望ましいでしょう（詳細は、第5章「ランゲージ」の「プロセス」に関する説明を参照してください）。

パスワード

新しいパスワード管理機能が備わりました。

ユーザモードへのアクセス権を持つグループを定義できます。

デフォルトユーザを作成できます。

セキュリティを改善したパスワードに変更または作成することができます。

注：4DWeb サーバがパスワードを管理する方法も変更されました。詳細は、第4章「Webサーバ」の「接続セキュリティ」を参照してください。

ユーザモードへのグループのアクセス

バージョン6.5では、グループでのユーザモードへのアクセスを定義することができ、カスタムモードからユーザモードへの切り替えを保護できるようになりました。

ユーザモードへのアクセス権を持っているグループの一員ではないユーザは、メニューを使っても、「option+f」キー（Macintoshの場合）のショートカットを使っても、カスタムモードからユーザモードへの移動ができません。ユーザが適切な権利を持たずにユーザモードにアクセスしようとするすると4Dは自動的に中断されます。

デザイナと管理者は、ユーザモードへのアクセス権を持つグループの明確な一員でなくても、常にユーザモードへのアクセスが可能です。

デザインモードへのアクセス権を持つユーザは、ユーザモードへのアクセス権を持つグループの明確な一員でなくても、常にユーザモードへのアクセスが可能です。

グループにユーザモードへのアクセス権を与えるには、

- 1 データベースプロパティで、2番目の「データ制御&アクセス権」タブをクリックする。
対応するダイアログが現れます。
- 2 「ユーザモード」ドロップダウンリスト内のグループを選択する。
4Dのパスワードエディタ内に前もってアクセスグループを定義しておく必要があります。
- 3 「OK」ボタンをクリックする。

デフォルトユーザ

データベースを使用するための「デフォルトユーザ」を定義することができます。このオプションが有効になると、のユーザがデータベースを開いたり接続する時、デフォルトユーザ以外の「パスワード」ダイアログが表示されません。さらに、デフォルトユーザにパスワードを設定していないと、「パスワード」ダイアログは表示されず、データベースは直接開きます。

各ユーザは、デフォルトユーザ用に定義されたアクセス権および制限を持ちます。

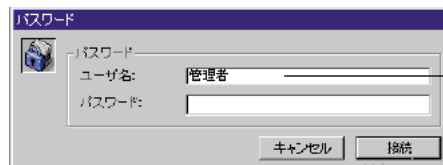
このオプションは、完全なデータ管理システムを保持しながらも、データベースへのアクセスを単純化しています。

デフォルトユーザを定義するには、

- 1 デザインモードで、パスワードエディタ内にユーザ（名前は重要ではありません）を作成する。
ユーザにパスワードを設定することができますが、必須ではありません。
- 2 このユーザに対するアクセス権と制限を選択する。
- 3 データベースプロパティで「データ制御&アクセス権」タブをクリックする。
- 4 「デフォルトユーザ」ドロップダウンリストで作成したユーザを選択する。
- 5 「OK」ボタンをクリックする。

データベースへのアクセスは、もはやカスタマイズされたものではありません。

デフォルトユーザにパスワードを設定すると、データベースが開かれた時にダイアログが現れて、デフォルトユーザのパスワードを入力しなければなりません。



デフォルトユーザに
定義された名前

デフォルトユーザにパスワードを設定していないと、上記のダイアログは現れません。

「パスワード」ダイアログを再表示する

例えば、デザイナまたは管理者としてデータベースに接続するために、4Dに強制的に標準「パスワード」ダイアログを表示させることができます。

「デフォルトユーザ」モードが起動された時に、「パスワード」ダイアログを再表示するには、

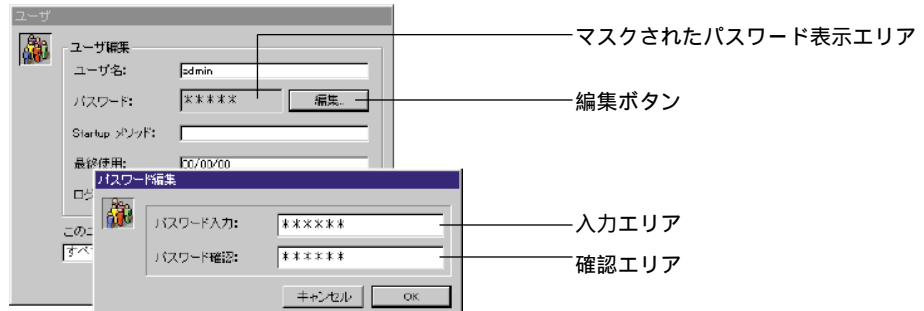
- 1 shiftキーを押しながらデータベースを開く（またはデータベースに接続する）。
ダイアログが現れて名前とパスワードを入力できるようになります。

パスワードの作成と変更

ユーザパスワードの作成と変更が改善されました。タイプミスを軽減するため新しいパスワードを2回入力しなければなりません。

パスワードの作成と変更を行うには、

- 1 ユーザ定義ダイアログで「編集...」ボタンをクリックする。
「入力」ダイアログが現れます。
- 2 入力エリアにパスワードを入力し、次に確認エリアにもう一度入力する。



- 3 「OK」ボタンを押します。
2つの入力が異なっていると4Dは警告音を鳴らしてダイアログが開いたままになります。

ピクチャライブラリ

ピクチャライブラリが提供する可能性はバージョン6.5で更に広がりました。今回の変更は主に下記の通りです。

ピクチャの管理と参照

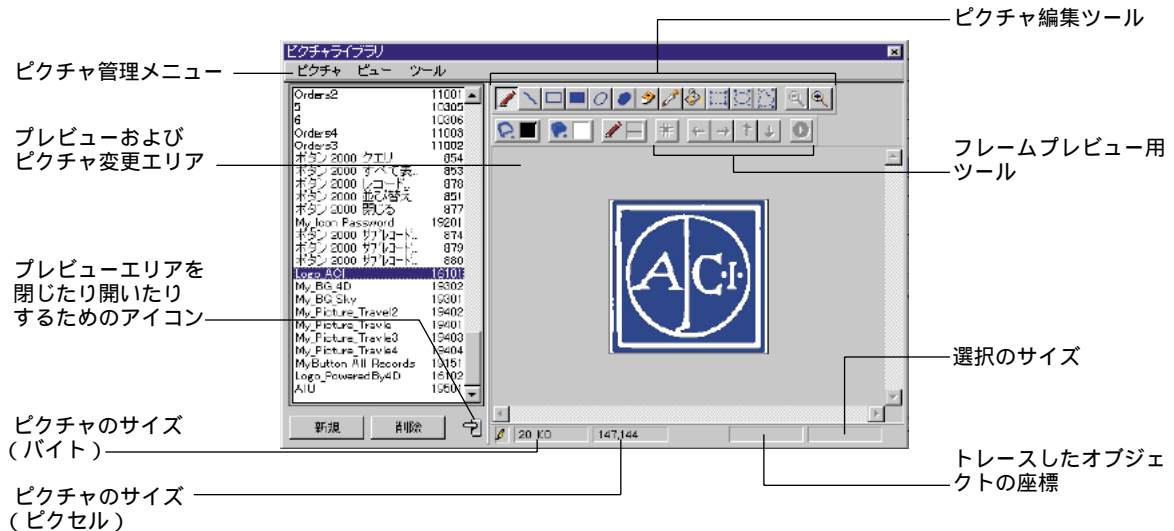
ライブラリ内でピクチャを直接作成し、変更する

ピクチャフレームの変更と作成

ピクチャの管理と参照

ピクチャライブラリはフローティングパレットで表示され、ライブラリを前面に保ちながら、異なるウインドウ（例えば、異なるフォーム）と同時に作業することができます。

ウインドウには、ピクチャ管理機能へのアクセスを可能にするメニューやピクチャの内容を編集するためのツールバーがあります。ピクチャリストの下の方にあるスイッチをクリックすることによって、ピクチャのビュー/エディットエリアを閉じたり開いたりすることができます。



ピクチャの追加

ピクチャをピクチャライブラリへ追加するには、3つの方法があります。

ピクチャファイルをインポートする。

ライブラリのピクチャメニューから「ピクチャ読み込み...」を選択します。現れる標準の「ファイルオープン」ダイアログ内で読み込むピクチャを選択します。

クリップボードからピクチャをコピーする。

この方法は、ピクチャライブラリの以前の機能に対応するものです。クリップボード内にピクチャをコピーし、次にライブラリのピクチャメニューから「クリップボード経由のピクチャ...」を選択します。

内容を描く空白ピクチャを作成する。

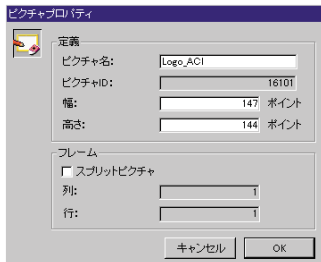
ピクチャライブラリには、洗練されたグラフィカルエディタがあります。これを使用するには、パレット内の「新規」ボタンをクリックするか、またはピクチャメニューから「新規ピクチャ...」を選択します。ピクチャの作成と変更に関する詳細は、この章の「ピクチャの作成と変更」を参照してください。



ピクチャのプロパティ

どのようにピクチャを追加しても、ピクチャのプロパティを定義するために下記のダイアログが現れます。

ピクチャ名：ピクチャのデフォルト名を変更できます。



ピクチャID：ピクチャにID番号を割り当てます。

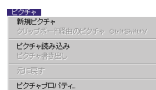
警告：ピクチャを作成した時にこの番号を定義できますが、これは後で変更することができません。

幅および高さ：ピクチャのサイズを定義します。これらの値は、（ファイルまたはクリップボードから）ピクチャがインポートされた時に前もって計算されています。ピクチャを分割する時（下記参照）値は各フレームのサイズになります。

フレームエリア：ボタンまたはピクチャメニューを作成するために、ピクチャのフレームの分割を定義できます。詳細は、この章の「フレームアシスタント」を参照してください。

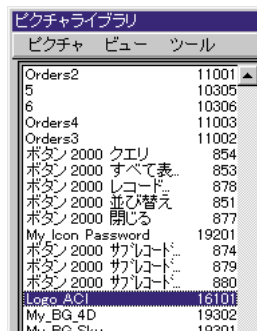
ピクチャを作成するには「OK」ボタンをクリックします。ピクチャのプロパティが定義されると、ピクチャを選択し、ライブラリのピクチャメニューから「ピクチャプロパティ...」を選択することで、いつでも変更することができます。

ピクチャを見る



ビューメニューを使用することによって、ピクチャライブラリ内のイメージをどのようにして見たいかを定義することができます。

デフォルトモード



ピクチャとして
見る













ピクチャの作成と変更

ピクチャライブラリにはペイントプログラムのようなツールバーがあり、ピクチャを描く、あるいは変更することができます。

ピクチャを作成、あるいは変更するには、ビュー/エディットエリアをクリックします。

ピクチャライブラリのエディタは、ビットマップモードで動作します。もちろん、ファイルまたはクリップボードから異なるタイプのピクチャ（例えばベクトルピクチャ）をインポートすることができ、その特性はデータベース内で使用する時にも維持されます。しかし、このピクチャをエディタ内で変更すると、ビットマップに変換され、変更されたピクチャが保存されると、その特性は失われます。インポートされたピクチャをライブラリ内で変更する時は、「警告」ダイアログが現れてピクチャがビットマップに変換されることを通知します。

ピクチャライブラリのグラフィックツールは次の通りです。

| アイコン | ツール | 機能 | オプションキー |
|---|---------------------|---------------------------|---|
|  | ペンツール | フリーハンドラインの描画。 | Alt(option)：カーソル上の色を採用します。 |
|  | ラインツール | 線の描画。 | shift：45度の角度で線を引きます。 |
|  | 長方形（空白）+ 長方形（塗りつぶし） | 長方形（空白）の描画。長方形（塗りつぶし）の描画。 | shift：正方形を描きます。 Ctrl(command)：長方形をその中心から描きます。 |
|  | 楕円（空白）+ 楕円（塗りつぶし） | 楕円（空白）の描画。楕円（塗りつぶし）の描画。 | shift：円を描きます。 Ctrl(command)：楕円形をその中心から描きます。 |
|  | 消しゴムツール | 現在の背景色を使って消去。 | |
|  | スポイトツール | 画面から色をサンプリングし、使用。 | |
|  | 塗りつぶしツール | 囲まれたエリアを指定した背景色で塗りつぶし。 | |
|  | 選択ツール | 画面の一部をドラッグで選択。 | |
|  | ズームツール | ピクチャの拡大/縮小 | |
|  | 描画色 + 背景色 | 線の色および背景色のメニュー | エディタのコンテキストメニューから変更可能です（Windowsではマウスの右ボタン、MacOSでは、controlキーを押しながらクリック）。 |
|  | 線の幅 | 線の幅のメニュー | |

注：他の既存の編集コマンド（コピー、ペーストなど）も、編集メニュー、4Dのツールバーまたは標準のキーボードショートカットなどから任意に使用できます。

変更の保存と取り消し

ピクチャに施された変更は、編集エリアの外側でクリックをするとすぐに（編集エリアがフォーカスを失うとすぐに）、自動的にライブラリ内に保存されます。

警告：一旦ピクチャが保存されると、いずれの変更も元に戻せません。

ピクチャを変更している間、4Dの編集メニュー（またはツールバー）から「取り消し」を選択するか、標準のショートカット（「Ctrlキー+Z」キー（Windowsの場合）または「command+Z」キー（Macintoshの場合））を使って、最後の変更を取り消すことができます。

また、ピクチャに施した全変更を、ライブラリのピクチャメニューから「元に戻す」を選択することによって取り消すこともできます。

フレームアシスタント

ピクチャライブラリには、「フレーム」を作成、あるいは変更することができる機能があります。フレームは、ピクチャボタンまたはピクチャメニューを作成するために使用します。これは、ピクチャを行または列に分割することで実現し、各セルが「フレーム」になります。4Dは指定されたパラメータに応じて、正しいフレームをピクチャボタンまたはピクチャメニュー内に表示させます（詳細は『4th Dimension デザインリファレンス』を参照してください）。

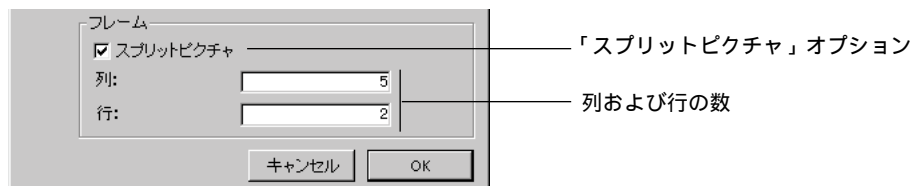
ピクチャを作成している時、またはその後も、フレームの並びを定義することができます。また、既に定義されたフレームの並びに、行、列またはフレームを挿入、あるいは削除することもできます。

フレームの並びの作成

既にピクチャライブラリに登録されたピクチャに対し、またはピクチャライブラリにピクチャを加える時に、フレームの並びを作成することができます。

両方の場合で、ピクチャが「ピクチャプロパティ」ダイアログ内でどのように分割されるかを定義することができます。ピクチャを作成する場合はダイアログが自動的に現れ、そうでない場合は、ピクチャを選び、ピクチャメニューから「ピクチャプロパティ...」を選択します。

フレームエリアでフレームの並びの行数と列数を定義することができます。フレームを作成するには、「スプリットピクチャオプション」をチェックしておく必要があります。



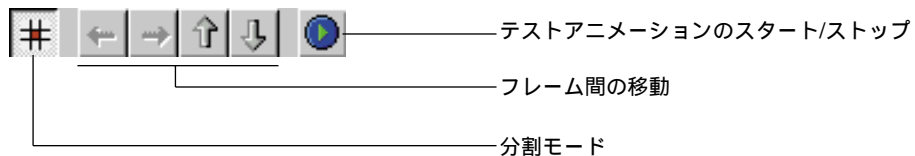
フレームのサイズ

フレームのサイズは4Dによって自動的に計算されます。フレームの並びを定義する時、「幅」と「高さ」のエリアが変更され、各フレームのサイズが表示されます。

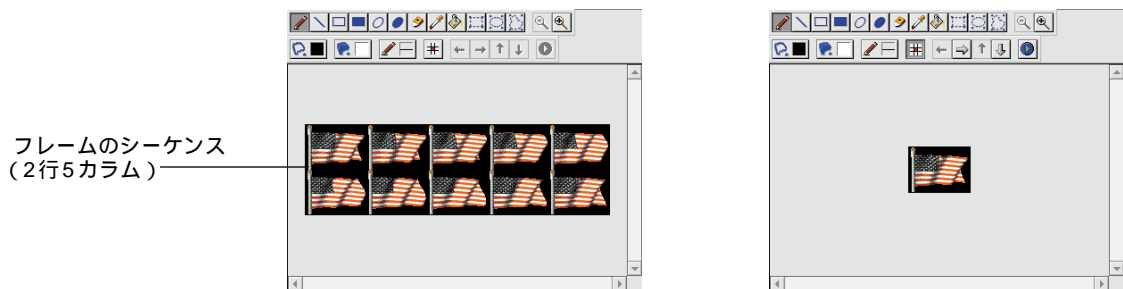
フレームのサイズを後で変更したい場合には、ピクチャのグローバルサイズについて心配することなく「幅」と「高さ」に新しい値を入力すれば、たとえフレームより大きくなっても、ピクチャが変形されることなく自動的に新しいサイズでセンタリングされます。新しいサイズが小さくなる場合は各フレームは縮小します。

フレームを見る

ピクチャライブラリのツールバーにある特定のツールを使って、各フレームをプレビューすることができます。



分割モードを実行すると、ピクチャを分割して左上から最初のフレームを表示します。その後は、「矢印」ボタンを使ってフレーム間を移動することができます。



すべてのフレームを自動的に表示させることもできます。この機能は、ピクチャボタンでアニメーションを作成したい場合に特に役に立つもので、アニメーションテストボタンをクリックします。フレームが連続して表示されます。テストを止めるには、アニメーションテストボタンをもう一度クリックします。

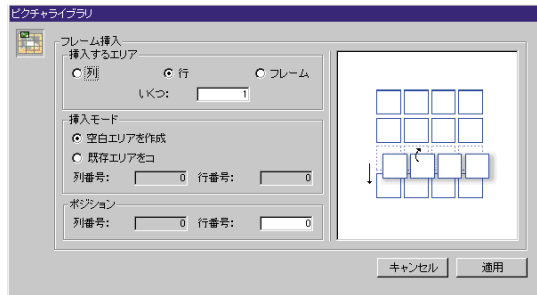
フレームの挿入と削除

以前に作成されたフレームの並びに、必要に応じてフレームを挿入、あるいは削除することができます。

フレームを挿入するには

- 1 変更したいフレームを選択する。
- 2 ピクチャライブラリのツールメニューから「挿入...」を選択します。

次のようなダイアログが現れます。



挿入したい列、行またはフレームを定義します。

ダイアログの右側のプレビューエリアには、選択された操作の例が表示されます。エレメントを挿入すると他のエレメントを移動させることに注意してください（どのエレメントも置き換わりません）。

3 挿入エリアに、挿入されたエレメントを空白にしたいか、それとも既存エレメントの内容を持たせるかを指示する。

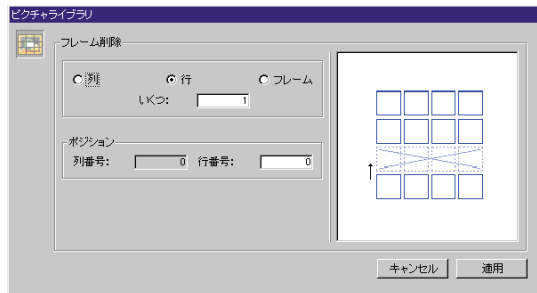
既存エレメントの内容を持たせる場合は、コピーするエレメントを指定しなくてはなりません。

注：最初の列および最初の行の番号は0です。

4 新しいエレメントを挿入したい位置を指示し、ダイアログを受け入れる。
エレメントは、これでピクチャ内に挿入されます。

フレームを削除するには、

1 ピクチャライブラリのツールメニューから「削除...」を選択する。
下記のダイアログが現れます。



削除したい列、行またはフレームを定義します。

ダイアログの右のプレビューエリアに、選択された操作の例が示されます。

2 削除したいエレメントの位置を指示し、「OK」ボタンを押します。

注：最初の列および最初の行の番号は0です。

ピクチャボタン/ポップアップメニューを挿入するショートカット

ピクチャがフレームの並びとして定義されると、フォーム内にピクチャボタンまたはポップアップメニューとして挿入するためのショートカットが使用できます。

ピクチャボタンを作成するには、ピクチャをライブラリからドラッグしてフォーム内にドロップします。

ピクチャポップアップメニューを作成するには、ピクチャをライブラリからドラッグして、shiftキーを押しながらフォーム内にドロップします。

注：フレームの並びとして定義されていないピクチャをドラッグ&ドロップすると、標準のピクチャとして挿入されます。

新しいプロパティリストでは、2つのタイプのピクチャを区別することができます。すなわち、ピクチャライブラリ（以前のバージョンのように、ライブラリ内のソースピクチャが変更された時にダイナミックに更新されるもの）と、スタティックピクチャ（ライブラリ内のソースピクチャとは関連付けられていないもの）です。詳細は、この章の「プロパティリスト」を参照してください。

イベントビューワ

バージョン6.5は、WindowsNTの「イベントビューワ」の利点を得ることができます。WindowsNTのこのサービスは、現在実行されているシステムおよびアプリケーションから来るメッセージを受け取って保存し、セッション管理ができるようにしています。

構成

この機能を使用するには、下記の条件が満たされていなければなりません。

4DアプリケーションがWindowsNTで実行されていること。

4DMSG.DLLファイルが、Windowsシステムフォルダ内のSystem32サブフォルダ内に
あること。

WindowsNTイベントログが既に開始されていること。

イベントビューワを見る

WindowsNTのスタートメニューで、「プログラム・管理ツール（共通）」を選び、次に「イベントビューワ」を選んでください。

ログメニューでアプリケーションを選択します。イベントビューアウインドウに、アプリケーションによって送られたメッセージが表示されます。

メッセージタイプ
アイコン

- 青 : 情報
- 黄色 : 警告
- 赤 : エラー

| 日付 | 時刻 | ソース | 分類 | イベントID | ユーザー | コンピュータ |
|----------|-------------|---------------|----|--------|------|-----------------|
| 99/07/05 | 午後 12:42:23 | 4th Dimension | なし | 3 | N/A | PC_LOCALIZATION |
| 99/07/05 | 午後 12:42:22 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/05 | 午前 9:23:12 | 4th Dimension | なし | 4 | N/A | PC_LOCALIZATION |
| 99/07/05 | 午前 9:21:33 | 4th Dimension | なし | 3 | N/A | PC_LOCALIZATION |
| 99/07/05 | 午前 9:21:31 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/04 | 午後 7:24:06 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/04 | 午後 4:05:58 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/04 | 午後 2:51:42 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/04 | 午後 2:51:16 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/04 | 午後 2:51:06 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/04 | 午後 2:49:52 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |
| 99/07/04 | 午後 2:49:28 | 4th Dimension | なし | 2 | N/A | PC_LOCALIZATION |

イベントを引き起こした
アプリケーション

メッセージの詳細を得るには、列をダブルクリックします。4th Dimensionがデータベースのスタートアップ、Webサーバなどに関する情報メッセージを送ります。

LOG EVENT コマンドを使用することによって、4Dデータベースからイベントビューア内に任意のメッセージを作成することができます。詳細は、第5章「ランゲージ」の「システム環境」を参照してください。

バージョン6.5のユーザモードでは、いくつかの機能やツールが変更されています。

データ読み込みおよびデータ書き出しの機能が新しくなりました。

ラベルエディタが修正されました。

メソッド実行のダイアログボックスに新しいオプションが加えられました。

データ読み込みとデータ書き出し

バージョン6.5では、「データ読み込み」および「データ書き出し」機能が改善されました。新しい特長としては、新しいエディタ、データ読み込みおよびデータ書き出しの手順を設定する新しいオプション、新しいファイルフォーマットおよび設定をファイルに保存し、読み出すことができます。

注：新しい「データ読み込み」および「データ書き出し」コマンドが、バージョン6.5に加えられました。第5章「ランゲージ」の「データ読み込みとデータ書き出し」を参照してください。

新しいエディタ

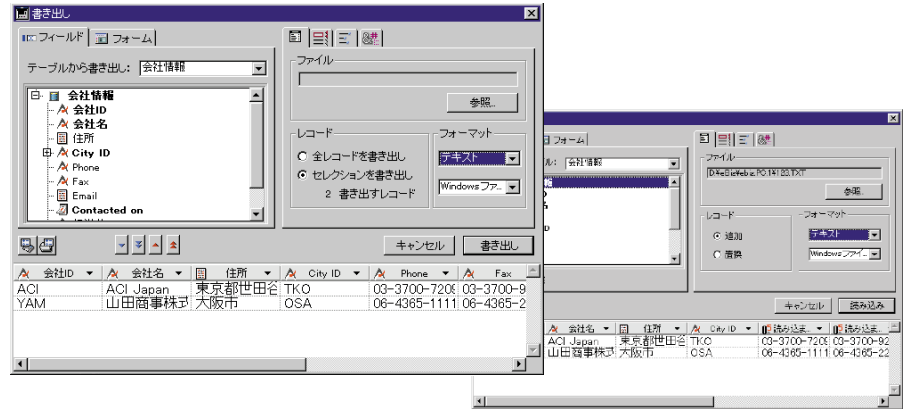
「データ読み込み」および「データ書き出し」は、新しいエディタを使用して実行します。エディタには4つのエリアがあります。

オプションエリア

テーブルおよび
フィールド選択
エリア

コマンドエリア

プレビューエリア



データ書き出しエディタ

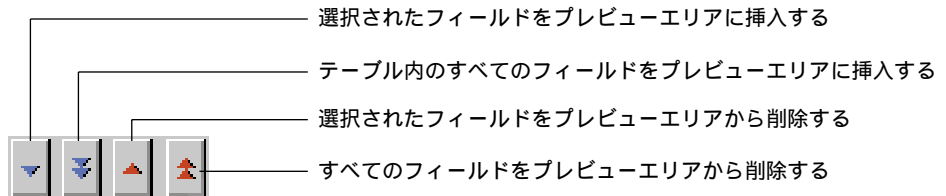
データ読み込みエディタ

テーブルおよびフィールド選択エリア

このエリアでは、データの読み込み、または書き出しをしたいフィールドを定義します。以前のバージョンのように、データの読み込み、または書き出しをしたいフィールドを手動で選択することも、4Dのフォームを使用することもできます。

エリアのメニューでテーブルを選択することができるので、前もって指定しておく必要はありません。

データ書き出し：書き出すフィールドを手動で選ぶには、階層リストの中のフィールド名をダブルクリックするか、またはそれをクリックした後に挿入ボタン（コマンドエリアにあります）をクリックします。エディタには、プレビューエリア内にフィールドの挿入や削除をする数多くのボタンがあります（データ書き出し用にフォームを使用しない場合のみ）。



データ読み込み：選択されたテーブル（デフォルトでは最初のテーブルが選択されます）のフィールドが階層リストとして参照できます。各入力の値はプレビューエリアに表示されます（詳細は、この章の「プレビューエリア」を参照してください）。

テーブル作成オプションを指定すると、データ読み込みエディタが新しく作成され、そのテーブルにデータを読み込むことができます。その際、必要なフィールドの数と各フィールドのタイプは読み込まれるデータに応じて決定されます。

作成されたフィールドのデータタイプは、プレビューエリア内のフィールドのタイトルバーをクリックして変更することができます。メニューには使用可能なタイプが表示されます。



データ読み込みを取り消す、あるいはオプションを外すと、テーブルは作成されず、ストラクチャは変更されません。

データ読み込み、またはデータ書き出し用にフォームを使用したい場合は、エリア上部にある「フォーム」タブをクリックし、テーブルのフォームリストから選ぶことができます。データの読み込み、または書き出しは、フォーム内で定義されているフィールドの入力順に行なわれます。入力可能な変数を含んだフォームを使用することもできます。変数ならびにフィールドに関連付けられたメソッドは、それぞれのデータが読み込み、または書き出される時に実行されます。しかし、これらのメソッドはプレビューエリアでは実行されないことに注意してください。

フィールドへの値（データ読み込みエディタ）



「フィールドへの値」
ボタン

「フィールドの値」ボタンは、フィールドが指定された順番ではなく、読み込みファイルの各データのタイプに応じてテーブル内のフィールドを「知的に」割り当てることを可能にします。読み込んだデータのタイプが推測され、次に対応する4Dフィールドに割り当てます。この解釈は下記のように行なわれます。

| 値の推定値 | 4Dのタイプ |
|-------------|--------|
| 数字のみ | 実数（数値） |
| True/False | ブール |
| 4D日付形式のいずれか | 日付 |
| 時間 | 時間 |
| その他 | 文字列 |

対応するフィールドの判断がつかない場合は、データ読み込みの値が「読み込まない」となります。このデータを読み込むには、プレビューエリア内でフィールドを手動で割り当てなければなりません。

注：このオプションは、データ読み込み用のフォームを使用している場合やテーブル作成チェックボックスを選択した場合は使用できません。

プレビューエリア

このエリアでは、データの読み込みまたは書き出しの内容を、データの並びとして見るすることができます。各カラムのサイズは、タイトルバーの境界線をマウスで選択してドラッグすることによって変更できます。

各カラムのタイトルバーは、データの読み込みまたは書き出されたフィールドのタイプ（アイコンによって）および名前を表わします。タイトルバーをクリックして下記の情報を得ることができます。

データ書き出し：書き出すテーブルのフィールドならびにリンクしているテーブルのフィールドのリストが現れ、書き出すフィールドを変更することができます。



タイトルバー内をマウスの右ボタンでクリック（Windowsの場合）またはControlキーを押しながらクリック（Macintoshの場合）すると、選択されたフィールドで使用可能な表示フォーマットのリストが現れ、書き出し時に使用されるフォーマットを定義できます。このオプションを変更しない場合は、デフォルトのフォーマットが使用されます（このオプションは、「フォーマットページ」内でも定義することができます）。

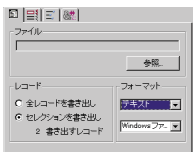
カラムを選択してdeleteキーを押すと、選択したカラムが書き出しフォーマットとなり、プレビューエリアから取り除かれます。

データ読み込み：読み込むテーブルのフィールドリストが現れ、データが読み込まれるフィールドを変更することができます。また、読み込まないオプションも選択することができます。この場合には、データのカラムにはデータが読み込まれません。

新しいオプション

「データ読み込み」および「データ書き出し」エディタは、「追加」タブを通してアクセス可能な新しいオプションを提供しています。タブの数（1から5）は、使用されているファイルのタイプおよび実行される操作のタイプによって変わります。

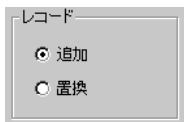
ファイルページ



「ファイル」エリアは、ファイル名とパスおよび選択された読み込みまたは書き出しファイルのタイプを示します。参照ボタンを使用すると、別のファイルを選択、あるいは作成できるようになります。

「レコード」エリアでは、カレントセクションの使用を指定することができます。

データ書き出し：選択されたテーブルの全レコード（「全レコードを書き出し」オプション）か、またはカレントセクション（「セクションを書き出し」オプション）かの、どちらを書き出したいかを指示します。選択された方法のレコード数が表示されます。



データ読み込み：読み込まれたレコードでカレントセクションを置き換えるのか（「置換」オプション）またはカレントセクションに加えて新しいセクションを作成するのか（「追加」オプション）を指示します。

「フォーマット」エリアでは、読み込みまたは書き出しファイル（テキストファイルのみ）のフォーマットを示すことができます。これらの設定に応じて、オプションページが表示されたりされなかったりします。新しいフォーマット固定長テキストおよび4th Dimensionは、後述の「新しいファイルフォーマット」で説明されています。

また、ファイルのタイプが、MacintoshなのかWindowsなのかを定義することもできます。このオプションは、デリミタ¹の定義に影響を及ぼします。

Macintosh ファイル：デリミタはMacintosh 標準のものが使われます（フィールドの終わり=tab、レコードの終わり=Return、ファイルの終わり=なし）。

Windows ファイル：デリミタはWindows 標準のものが使われます（フィールドの終わり=tab、レコードの終わり=Return+ラインフィード、ファイルの終わり=なし）。4th Dimension の場合、Macintosh/Windows 変換フィルタが用いられます（内部フィルタは、確実に4th Dimension のマルチプラットフォームコンパティビリティであることを覚えておいてください。4th Dimension はMacintosh ASCII フォーマットでデータを保存します）。

ヘッダページ

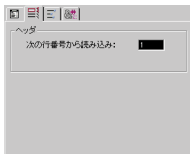


このページでは、データ書き出し用にファイルヘッダを指定します。

タイトル：このオプションにより、書き出しファイルにタイトルを入力することができます。チェックボックスの下エリアにタイトルを入力します。

列タイトル：このオプションは、フィールド名であるカラムのタイトルを書き出します。

行番号：このオプションは、書き出されたレコードである各行に番号を付けます。番号付けは1から始まり、インクリメントします。

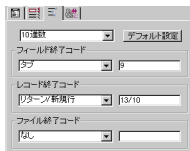


テキストファイルから読み込むには、2つのオプションがあります。

次の行番号から読み込み：このオプションにより、読み込み開始行を指定できます。これは、読み込みファイルがフォーマットされていない行（タイトルや日付など）で始まる場合に特に役立ちます。なぜならば、読み込むカラムは最初の行のフォーマットによって計算されるためです。

フィールド名を列タイトルにする：このオプションは、カラムタイトルをフィールド名として使うように指示するのでもです（テーブル作成オプションを選択した場合）。

デリミタページ



このページは、データ書き出しまたは読み込みファイルで使用され、デリミタを定義するものです（読み込みは、テキストファイルでなければなりません）。

標準のデリミタとは別に、新しいタイプデリミタの「レコード終了コード」があります。このデリミタは、ファイルを特定のアプリケーションと交換する場合に必要です。

デリミタを変更するには、各エリアの左のスクロールダウンメニューを使用します。これらのメニューには、各デリミタタイプに最も頻繁に使用される値が含まれています。

右側のエリアにも直接デリミタを入力できます。

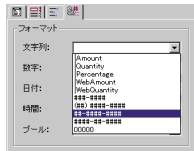
1. このオプションは、デリミタページで値の指定を可能にするものです。また、このページで直接変更することも可能です。



ページのトップにあるメニューはデリミタを文字、10進コード（デフォルト）、16進コードのどのフォーマットで表示するか指定できます。

注：デリミタは、「ファイル」ページでドキュメントのタイプが変更された時、デフォルト値に初期化されます。

フォーマットページ



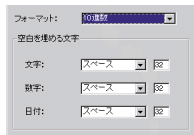
このページで、読み込みまたは書き出された値（テキストファイルのみ）のフォーマットを定義できます。デフォルトでは、4Dの標準フォーマットが使用されます。

データ書き出し：文字列、数値、日付、時間、ブールのフォーマットを定義できます。これを行うには、プレビューエリア内でフォーマットするカラムを選択します。フィールドタイプにより、対応するコンボボックスがフォーマットページ内で選択できるようになります。次に4Dフォーマットを選択またはカスタムフォーマットを入力します。

注：コンテキストメニューを使用してもカラムのフォーマットを定義することができます（詳細は、この章の「プレビューエリア」を参照してください）。

データ読み込み：ブール用のフォーマットを指定することができます。指定の仕方は「真の値;偽の値」です。値がBlack（真）およびWhite（偽）であるブールフィールドを読み込むには「Black;White」と入力します。ドロップダウンリストは、最も一般的に使用されているブールフォーマットを表示します。

埋め文字ページ



このページは、ファイルフォーマットが「固定長テキスト」の場合にのみ現れます。値のタイプに応じて使用する補充文字を定義することができます。

補充文字を使用できるデータタイプは、「文字列/ブール（ブールは文字として書き出されます）」、「数字（整数、倍長整数および実数）」および「日付/時間」の3つです。

メニューは、最も頻繁に使用される値を表示します。右の入力エリアは、選択された文字コードを表示します。他の文字コードを使用したい場合には、入力エリアに直接入力してください。

ページのトップにあるメニューは、補充文字を文字、10進コード（デフォルト）、16進コードのどのフォーマットで表示するか指定できます。

新しいファイルフォーマット

バージョン6.5では、データを2つの新しいファイルフォーマット、4th Dimensionおよび固定長テキストでデータ読み込み、またはデータ書き出しができます。

4th Dimension フォーマット

4th Dimension 専用の、この新しいファイルフォーマットは、異なる4Dデータベース間で簡単にレコードの交換ができるようにしたものです。さらに、このフォーマットでは、すべての4Dのデータタイプを交換することができます。ピクチャとBLOBフィールドも読み込みおよび書き出すことができます。書き出しファイルの拡張子は「.4IE」です。

固定長テキストフォーマット

固定長テキストフォーマットの主な特徴は固定長の値を使用するということで、それぞれのデータカラム内では同じ文字数でなければならないということです。もちろん、名前のように値が同じ文字数であることは不可能です。したがって、補充文字が足りない部分を埋めるために使用されます。各データタイプ用の補充文字を使用することもデフォルト文字を使用することも可能です（詳細は、この章の「埋め文字ページ」を参照してください）。しかし、値が定義された数よりも長い文字数の場合、長い部分は切り捨てられます。

データ書き出し：「固定長テキスト」フォーマットを選ぶと、プレビューエリアは各カラムの文字数を表示します。デフォルトでは、文字数は次の方法で設定されます。テキストフィールドは80文字で、文字列フィールドはストラクチャで定義された文字数と等しく、数値フィールドは10文字となります。これらのデフォルト値は、入力エリアに新しい値を入力するか、または手動で各カラムのサイズを変更することによって変えることができます。

補充文字は、文字フィールドの最後および数値フィールドの最初に加えられます。

データ読み込み：フォーマットが「固定長テキスト」であるファイルを読み込む時は、常にプレビューエリアに各カラム用の文字数を表示します。

データ読み込み、データ書き出しの設定

データ読み込みおよびデータ書き出しのダイアログで、各設定を設定ファイルに保存し、読み込むことができます。

データ読み込みまたはデータ書き出しの設定を保存し、読み込みするには、コマンドエリアの対応するボタンをクリックします。

データ読み込みおよびデータ書き出しの設定ファイルの拡張子は、Windowsでは「.4SI」（Macintoshでのファイルタイプは「4DSI」）です。データ読み込みまたはデータ書き出しのエディタで定義されたすべての設定を保存します。

ファイル名とそのアクセスパス

選択されたテーブルとフィールドならびに、選択されたフォームの名前

データ読み込みおよびデータ書き出しのオプション（ファイルタイプやデリミタなど）

データ読み込みまたはデータ書き出しをできるだけ自動化させることができます。設定ファイルの使用は、新しいデータ読み込み/データ書き出しコマンドで特に役立ちます（詳細は、第5章「ランゲージ」の「データ読み込みとデータ書き出し」を参照してください）。

注：データ読み込み/データ書き出し設定ファイルは、フィルタを保存しません。これらがシリアル通信に関係しているかもしれないからです。必要であれば、操作を行なう前にデータ読み込みまたはデータ書き出しフィルタをロードする必要があります。

ラベルエディタ

ラベルエディタで、フィールドを連結するには下記の方法があります。

2つのフィールドを単にドラッグ&ドロップで連結すると、使用されるセパレータは4Dのリソースで定義されたものになります。デフォルトでは「スペース」が使用されます。

2つのフィールドを「return」コードをセパレータとして連結することができます。これを行うには、shiftキーを押しながら第2フィールドをドラッグ&ドロップします。

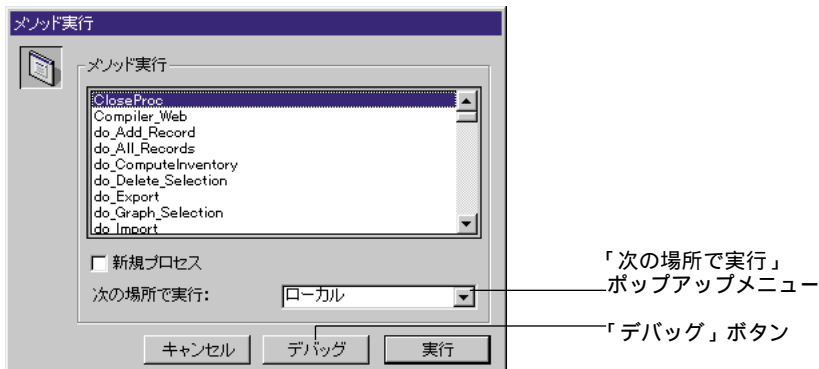
ラベルが印刷されている時に第2フィールドが空の場合は、その位置は削除されて詰められます。フィールドに改行付きのテキストが含まれている場合は改行が考慮されます。

例えば、多くの重ね合わされたフィールド([Clients]Address1、[Clients]Address2など)を使用して、空白行を作らずに1つの連続した住所で住所ラベルを作成することができます。

注：連結されたフィールドのグループを、それらが含む行数に合わせて自動的にサイズを変更することができます。これを行うには、グループの右下にあるハンドル上で、WindowsではCtrlキーを押しながらクリック、Macintoshではcommandキーを押しながらクリックします。

メソッドの実行

メソッド実行のダイアログには2つのオプションがあります。



デバッグ：このボタンは、トレースモードでメソッドを実行します（詳細は、第5章「ランゲージ」の「トレースモードでのメソッドの実行」を参照してください）。

「次の場所で実行」メニュー：このオプションは、指定されたメソッドを「4D Server上」が選択された場合はサーバ上で（ストアードプロシージャ）、そうでない場合は選択した他のクライアント上で、実行することができます。

注：「4D Server上」オプションは、以前のバージョンにあった「4D Server上で実行」オプションに代わるものです。

他のクライアントワークステーションがメニューに現れ、メソッドを実行できるようにするには、前もって登録されていなければなりません（詳細は、**REGISTER CLIENT**ルーチンの説明、ならびに第2章「デザインモード」の「スタートアップ時にクライアントを登録する」（4D Serverのみ）を参照してください）。

注：デフォルトでは、「ローカル」オプションが選択されます。4Dのシングルバージョンではこのオプションのみです。

バージョン6.5では、統合されたWebサーバに多くの新しい機能と改善が備わりました。新しい機能は次のエリアに設けられています。

接続セキュリティ：接続セキュリティが、新しいオプションと新しいデータベースメソッド「On Web Authentication」により強化されました。「一般Webユーザ」オプションは、データベース内のアクセス管理を単純化します。さらに、デフォルトHTMLの設定とルートフォルダを指定することで、ディスク上のファイルへのアクセスを制御できます。

カスタマイズセッション：Webサーバ用に多くのパラメータが付加されました。デフォルトホームページを定義する、データ入力を管理するためのJavaScriptを使用する、新しいコンテキストモードを使用する、HTMLがどのように変換されるかを定義するといったことが可能です。バージョン6.5は、スタティックホームページ、ピクチャおよびスタイルシート用にキャッシュを使用することができます。また、サーバがHTTPリクエストを許可するIPアドレスを設定できます。

Webサイトの機能に関する情報：特定のURLは、Webサイトがどのように機能しているかについての情報を得ることを可能にしました。また、Webサイトへの接続に関する情報を含んだログファイルを作成することも可能です。

HTMLサポート：4D フォームのHTMLへの変換、一般的なHTMLのサポートが、改善されました。

非コンテキストモード：非コンテキストモードは、4D 6.5 Webサーバの主要な改革です。コンテキストの管理から開放し、4D 6.5 Webサーバを標準のHTTPサーバにします。このモードでは、「セミダイナミック」ページを使用することで4D能力の利点を発揮できます。

注：バージョン6.5のプログラミング言語である「Webサーバ」テーマ内には、沢山のコマンドが追加、あるいは変更されました。これらのコマンドの情報は、第5章「ランゲージ」を参照してください。

接続セキュリティ

4D Web サーバの接続セキュリティは、下記の要素によって強化されました。

Web アクセス用の新しいパスワード管理システム

新しいデータベースメソッドの On Web Authentication

「一般 Web ユーザ」の定義

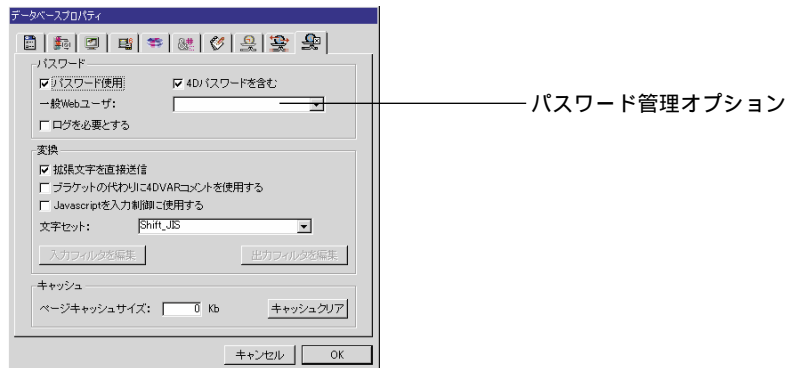
デフォルト HTML、ルートフォルダの設定

パスワードオプション

「データベースプロパティ」ダイアログ内で、Web サーバに適用したいアクセス管理システムを指定することができます。

パスワードシステムを定義するには、

- 1 「データベースプロパティ」ダイアログ内で、「Web サーバ II」タブをクリックする。
次のウィンドウが現れます。



「パスワード使用」と「4D パスワードを含む」の2つのオプションが選択できます。「4D パスワードを含む」のチェックボックスは「パスワード使用」が選択された場合にのみ指定できます。

パスワード使用：Web サーバのパスワードシステムを起動させます。接続時にブラウザ上にダイアログボックスが現れ、名前とパスワードが入力できます。

名前とパスワードならびに接続パラメータ(IP アドレスおよびポート、URL、...)は、On Web Authentication データベースメソッドに渡され、独自のパスワードシステムの構築など、必要な処理ができるようになります。

注：この場合、On Web Authentication データベースメソッドが存在しなければ接続は拒絶されます。

4D パスワードを含む：独自のパスワードシステムの代わり、またはそれに付加するものとして、4D で定義されているデータベースパスワードシステムを使用することができます。

Web アクセス管理システムは、これらのオプションと On Web Authentication データベースメソッドの組み合わせの結果です。このデータベースメソッドについての詳細は、次項の「On Web Authentication データベースメソッド」を参照してください。4D Web サーバのアクセスシステムについての詳細は、この章の「4D Web サーバのアクセスシステム」を参照してください。

2 オプションを定義して「OK」ボタンをクリックします。

On Web Authentication データベースメソッド

新しいデータベースメソッド「On Web Authentication」がバージョン 6.5 に追加されました。このデータベースメソッドは、Web サーバへのアクセス管理の役目があります。

このメソッドには、On Web Connection データベースメソッドのように、6 つの引数 \$1、\$2、\$3、\$4、\$5、\$6 が渡されます（詳細は、この章の「On Web Connection データベースメソッドの変更」を参照してください）。

これらの引数の内容は次の通りです。

| 引数 | タイプ | 内容 |
|-----|------|---------------------|
| \$1 | テキスト | URL |
| \$2 | テキスト | HTTP ヘッダ |
| \$3 | テキスト | Web クライアントの IP アドレス |
| \$4 | テキスト | サーバの IP アドレス |
| \$5 | テキスト | ユーザ名 |
| \$6 | テキスト | パスワード |

注：On Web Authentication データベースメソッドの全引数が有効とは限りません。データベースメソッドが受け取った情報は、「データベースプロパティ」ダイアログ内で設定したオプションに基づきます（詳細は、次項の「4D Web サーバのアクセスシステム」を参照してください）。

\$1 および \$2 に関する詳細は、『4th Dimension ランゲージリファレンス』の「On Web Connection データベースメソッド」を参照してください。

\$3 には、ブラウザが動作しているマシンの IP アドレスが渡されます。この情報は、イントラネット接続とインターネット接続を区別できるようにするものです。

\$4 には、4D Web サーバの IP アドレスが渡されます。バージョン 6.5 は、複数の IP アドレスを持つマシンでマルチホーミングを可能にしています。詳細は、この章の「HTTP リクエスト IP プログラムの定義」を参照してください。

\$5および\$6には、ユーザによって入力されたユーザ名とパスワードが渡されます。「パスワード入力」ダイアログは、「データベースプロパティ」ダイアログ内で「パスワード使用」オプションが選択された場合、各接続時に表示されます。

On Web Authentication データベースメソッドはブールタイプの関数値を返します。

接続を受け入れる場合は、\$0にTrueを設定します。

接続を拒否する場合は、\$0がFalseを設定します。

On Web Connection データベースメソッドは、On Web Authenticationによって接続が受け入れられた場合にのみ実行されます。

On Web Connection データベースメソッド内では、いかなるインタフェースエレメント (ALERTやDIALOGなど) も呼び出してはなりません。呼び出した場合、操作は中断され、接続は拒絶されます。データベースメソッドが実行されている間にエラーが発生した場合も同様です。

On Web Authentication データベースメソッドの代表的な例を示します。

```
C_TEXT($5;$6;$3;$4)
C_TEXT($user;$password;$BrowserIP;$ServerIP)
C_BOOLEAN($4Duser)
ARRAY TEXT($users;0)
ARRAY TEXT($nums;0)
C_LONGINT($upos)
C_BOOLEAN($0)
$0:=False
$user:=$5
$password:=$6
$BrowserIP:=$3
$ServerIP:=$4 `
`セキュリティのため、@を含む名前を拒否
IF(WithWildcard($user) | WithWildcard($password))
    $0:=False
    `WithWildcard メソッドは下記に示します
Else
    `4D ユーザのチェック
    GET USER LIST($users;$nums)
    $upos:=Find in array($users;$user)
    If($upos > 0)
        $4Duser:=Not(Is user deleted($nums{$upos}))
    Else
        $4Duser:=False
    End if
    If (Not($4Duser))
        `4D ユーザではありません。Web ユーザテーブルを参照
        QUERY([WebUsers];[WebUsers]User=$user;*)
        QUERY([WebUsers]; & [WebUsers]Password=$password)
```



```

    $0:=(Records in selection([WebUsers]) = 1)
Else
    $0:=True
End if
End if
`イントラネット接続ですか?
If (Substring($BrowserIP;1;7) # "192.100.")
    $0:=False
End if

WithWildcard メソッド :
C_INTEGER($i)
C_BOOLEAN($0)
C_TEXT($1)
$0:=False
For($i;1;Length($1))
    If (Ascii(Substring($1;$i;1)) = Ascii("@"))
        $0:=True
    End if
End for

```

4 D Web サーバのアクセスシステム

4D Web サーバへの接続を判断するシステムは、2つのパラメータの組み合わせによりま

す。
「データベースプロパティ」ダイアログ内のWebパスワードオプション（「パスワードオプション」を参照）

On Web Authentication データベースメソッドの存在（「On Web Authentication データベースメソッド」を参照）

オプションが何も指定されていない場合：

On Web Authentication が存在している場合、\$1と\$2以外は、\$3と\$4にブラウザとサーバのIPアドレスが渡され、\$5と\$6のユーザ名とパスワードは空白のまま実行されます。この場合、ブラウザのIPアドレスまたはサーバのIPアドレスを使用して接続を判断することができます。

On Web Authentication が存在しない場合、接続は自動的に受け入れられます。

「パスワード使用」オプションが指定されて、「4Dパスワードを含む」オプションは指定されていない場合：

On Web Authentication が存在している場合、すべての引数が渡されます。したがって、ユーザ名、パスワード、ブラウザまたはサーバのIPアドレスに応じて、接続をより細かく判断することができます。

On Web Authentication が存在しない場合、接続は自動的に拒否されて、Authentication メソッドが存在しないことを示すメッセージがブラウザに送られます。¹

「パスワード使用」および「4D パスワードを含む」オプションが指定されてる場合

注：以前のバージョンで作成されたデータベースは、オプションがデフォルトのまままで開かれます。

ブラウザによって送られたユーザ名が4Dのユーザとして登録されており、パスワードが正しければ、接続は受け入れられます。²

パスワードが誤っていれば、接続は拒否されます。

ブラウザによって送られたユーザ名が4Dのユーザとして登録されていない場合、2つの可能性があります。

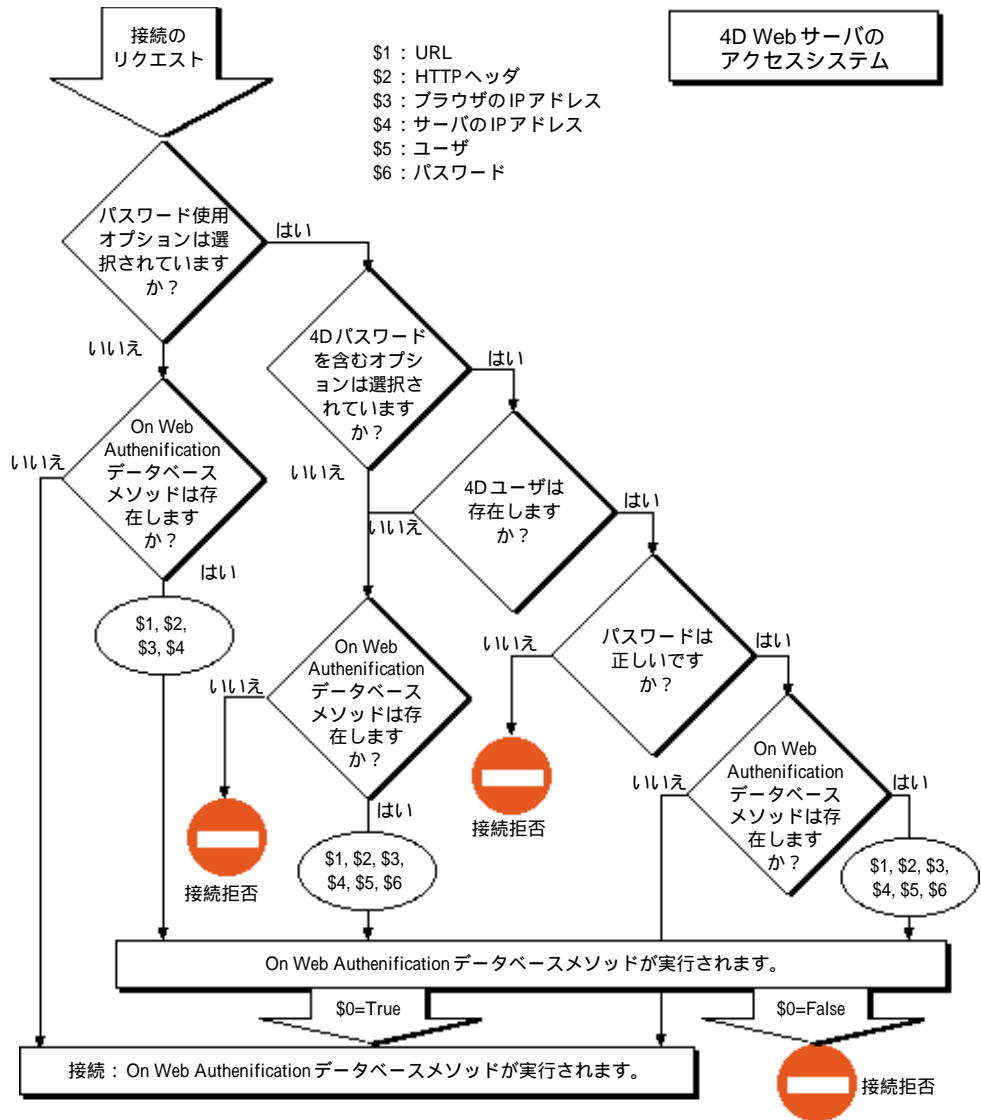
On Web Authentication が存在している場合、引数\$1、\$2、\$3、\$4、\$5、\$6が渡されるので、ユーザ名、パスワード、ブラウザまたはWebサーバのIPアドレスを使用して接続を判断することができます。

On Web Authentication が存在しない場合、接続は拒否されます。

1. ブラウザによって送られたユーザ名が空白で、On Web Authentication データベースメソッドが存在しない場合、「パスワード」ダイアログがブラウザに送られます。

2. ブラウザによって送られたユーザ名が4Dのユーザとして登録されている場合、引数\$6（ユーザのパスワード）へは、セキュリティの関係上何も渡されません。

4D Web サーバのアクセスシステムは、次の図のようにまとめられます。



ロボットに関するセキュリティ注意事項

特定のロボット（query engines、spiders...）は、Web サーバやスタティックホームページを閲覧していきます。ロボットにすべてのサイトをアクセスできるようにさせたい場合、彼らにどのURLへのアクセスを許さないのかを定義できます。

これを行うには、ROBOTS.TXT ファイルをサーバのルートに置きます。このファイルは下記の方法で構成されていなければなりません。

```
User-Agent : <name>  
Disallow : <URL> or <beginning of the URL>
```

例：

```
User-Agent:*  
Disallow:/4D  
Disallow:/%23%23  
Disallow:/GIFS/
```

「User-Agent:*」は、すべてのロボットに影響することを意味します。

「Disallow:/4D」は、/4D で始まる URL にロボットがアクセスできないことを意味します。

「Disallow:/%23%23」は、/%23%23 で始まる URL にロボットがアクセスできないことを意味します。

「Disallow:/GIFS/」は、/GIFS/ フォルダまたはそのサブフォルダにロボットがアクセスできないことを意味します。

```
他の例として、  
User-Agent:*  
Disallow:/
```

この場合、ロボットはサイト全体へのアクセスが許されません。

一般 Web ユーザ

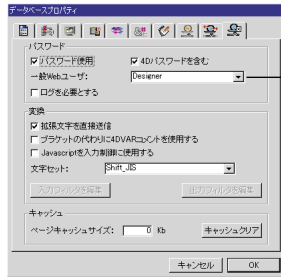
4D パスワードテーブルに定義されたユーザを「一般 Web ユーザ」として指定することができます。この場合、データベースに接続する Web ユーザは、この一般ユーザに設定されたアクセス権と制限が適応されます。したがって、データベースの異なる部分への閲覧を簡単に管理できます。

注：Web ユーザがデータベースの異なる部分（テーブルやメニューなど）へのアクセス制限を受けるこのオプションを、パスワードシステムで管理される Web サーバの接続管理システムと混同しないようにしてください（スタティックに関する説明を参照してください）。

一般 Web ユーザを定義するには、

- 1 デザインモードで、パスワードエディタ内にユーザを作成する。
必要であれば、ユーザにパスワードを設定できます。
- 2 各エディタで、このユーザのアクセス（可能または禁止）を設定する。
- 3 「データベースプロパティ」ダイアログで、「Web サーバ II」タブをクリックする。

次のようなページが現れます。



4Dのパスワードテーブルで定義されたユーザのリスト

デフォルトでは、一般ユーザはデザイナーであり、Webユーザはデータベース全体へのアクセス権を持ちます。

4 「一般 Web ユーザ」リスト内のユーザを選ぶ。

5 「OK」ボタンをクリックする。

データベースへの接続を許されたすべての Web ユーザは、この一般 Web ユーザに関連したアクセス権およびアクセスの制限が適応されます（「4D パスワードを含む」オプションが選択され、接続するユーザが4D パスワードテーブル内に存在しない場合を除きます。下記を参照してください）。

Web パスワードシステムとの相互作用

パスワード使用オプションは、一般の Web ユーザがどのように操作するかには影響を与えません。このオプションの状態によらず、「一般 Web ユーザ」に関連したアクセス権と制限は、データベースに接続することを許されたすべての Web ユーザに適用されます。

しかし、4D パスワードを含むオプションが選択された場合は、下記の2つの可能性があります。

ユーザ名とパスワードが4Dのユーザとして登録されていない場合、On Web Authentication データベースメソッドによって接続が受け入れられると、一般 Web ユーザのアクセス権が適用されます。

ユーザの名前とパスワードが4Dのユーザとして登録されている場合には、「一般 Web ユーザ」のパラメータは無視され、ユーザ自身のアクセス権で接続します。

デフォルト HTML ルートフォルダを定義する

データベースプロパティの、この新しいオプションは、4D がブラウザに送るスタイルシート HTML ファイルとピクチャが存在するフォルダの定義を可能にします。

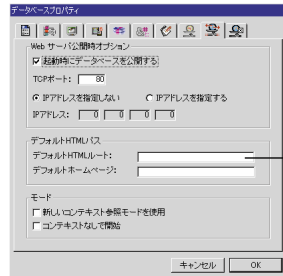
更に、HTML ルートフォルダより上位の階層へのアクセスができないように定義したことになります。

このアクセス制限は、Web ブラウザに送られる URL ならびに **SEND HTML FILE** のような 4D の Web サーバコマンドに適用されます。ブラウザによって送られた URL または 4D コマンドが、HTML ルートフォルダの上位にあるファイルにアクセスしようとした場合に、ファイルが発見できなかったことを示すエラーが返されます。

デフォルトでは、HTMLルートフォルダは、データベースのストラクチャファイルがあるフォルダです。この場合は、アクセス制限が何もないために注意が必要です（ユーザはすべてのボリュームにアクセス可能です）。

デフォルトHTMLルートフォルダを定義するには、

- 1 「データベースプロパティ」ダイアログで、「Webサーバ」タブをクリックする。
下記のページが現れます。



デフォルトのHTMLルートフォルダ
入力エリア

- 2 「デフォルトHTMLルート」入力エリアに、定義したいフォルダのアクセスパスを入力する。

このダイアログボックスに入力するアクセスパスは、データベースのストラクチャファイルがあるフォルダを起点に指定します。4D Webサーバはデータベースのマルチプラットフォームの互換性を保つため、アクセスパスを記述するための特別な書き込み規約を使用します。構文のルールは下記の通りです。

フォルダはスラッシュ (/) で分離すること

アクセスパスはスラッシュ (/) で終了すること

フォルダ階層の中で1レベル「上がる」には、ピリオドを2つ (..) を、フォルダ名の前に入力すること

アクセスパスはスラッシュ (/) で始めてはならないこと（HTMLルートフォルダをデータベースのフォルダにしたい場合を除きます。下記を参照してください）

例えば、HTMLルートフォルダを「4Ddatabase」フォルダ内の「Web」サブフォルダにしたい場合は、「4Ddatabase/Web/」を入力します。HTMLルートフォルダをデータベースフォルダにしたいが、上のフォルダへのアクセスを禁止したい場合は、スラッシュ (/) をエリア内に入力します。全ボリュームへアクセスを可能にするには、「デフォルトHTMLルート」エリアを空にします。

- 3 「OK」ボタンをクリックする。

注：HTMLルートフォルダが、「データベースプロパティ」ダイアログで変更された場合、キャッシュはクリアされ、アクセスが制限されているファイルを保存しないようになります。

データベースのプロパティと SET HTML ROOT

一旦HTMLルートフォルダが「データベースプロパティ」ダイアログで指定されると、**SET HTML ROOT** コマンドを使って変更することができますが、変更は現在のWebプロセスのみに適用され、HTMLページのキャッシュはクリアされます。

しかし、**SET HTML ROOT** コマンドでの設定は、データベースプロパティに定義されているデフォルトHTMLルートフォルダを考慮します。

「データベースプロパティ」ダイアログで定義されたフォルダが「WebPages/」の時、**SET HTML ROOT** ("Folder")を実行すると、デフォルトのHTMLルートフォルダは「WebPages/Folder/」になります。また、この場合「WebPages」フォルダの上位にあるフォルダへのアクセス制限は維持され、「WebPages」フォルダには設定されません。

注：Webサーバが非コンテキストモードの時は、**SET HTML ROOT** コマンドは何も行ないません（詳細は、この章の「非コンテキストモード」を参照してください）。

Web セッションのカスタマイズ

バージョン6.5は、以前にも増して、ユーザの要求に応じたWebセッションのカスタマイズができるようにしました。下記のオプションがあります。

- デフォルトホームページの定義
- データ入力管理でのJavaScriptの使用
- コンテキストモードの変更
- HTML変換の設定とWebフィルタの定義
- スタティックホームページのキャッシュ
- サーバがHTTPリクエストを許可するIPアドレスの定義

デフォルトホームページの定義

データベースに接続するすべてのWebユーザ用にデフォルトホームページを定義することができます。このページは、スタティックまたはセミダイナミックにすることができます（セミダイナミックに関しては、この章の「セミダイナミックページ」を参照してください）。

デフォルトホームページを指定すると、このページはWebセッションにコンテキストまたは非コンテキストのどのモードが定義されていても、接続する各ブラウザに送られます。コンテキストモードの場合、デフォルトホームページを指定していると、以前のバージョンとは異なり、メニューバーの内容がブラウザに送られません（詳細は、この章の「スタートアップ時に非コンテキストモードを定義する」を参照してください）。

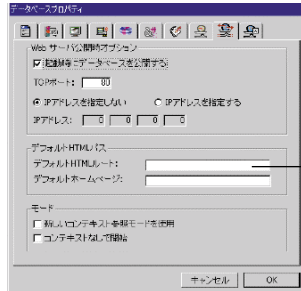
デフォルトでは何のホームページも定義されません。デフォルトホームページを指定しないと、Webサーバの動作はスタートアップモードによって異なります。

Webサーバがコンテキストモード（デフォルト）でスタートアップすると、以前のバージョンのように現在のメニューバー（デフォルトではメニューバーNo.1）がホームページとして送られます。

Webサーバが非コンテキストモードでスタートアップすると、On Web Connection データベースメソッドが呼び出されます。リクエストを手順を追って処理することも可能です。

デフォルトホームページを定義するには、

- 1 「データベースプロパティ」ダイアログで、「WebサーバI」タブをクリックする。
下記のページが現れます。



デフォルトホームページ
入力エリア

- 2 「デフォルトホームページ」入力エリアに、ホームページへのアクセスパスを入力する。
このダイアログボックスに入力するアクセスパスは、データベースのストラクチャファイルがあるフォルダを起点に指定します。4D Webサーバはデータベースのマルチプラットフォームの互換性を保つため、アクセスパスを記述するための特別な書き込み規約を使用します。構文のルールは下記の通りです。

フォルダはスラッシュ (/) で分離すること

アクセスパスはスラッシュ (/) で終了しないこと

フォルダ階層の中で1レベル「上がる」には、ピリオド2つ (..) を、フォルダ名の前に入力すること

アクセスパスはスラッシュ (/) で始めてはならないこと

例えば、デフォルトホームページを「MyHome.htm」として、それが「Web」フォルダ内にある場合は「Web/MyHome.htm」を入力します。

- 3 「OK」ボタンをクリックする。

注：各Webプロセス用のデフォルトホームページも、ルーチン**SET HOME PAGE**を使用して定義することができます（第5章「ランゲージ」の「SET HOME PAGE」を参照してください）。

データ入力管理に JavaScript を使用

バージョン6.5では、データ入力管理の一部を、JavaScriptを使用して自動的にブラウザ上で行なうことができます。

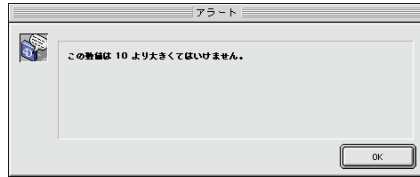
ブラウザ上で適用されるデータ入力管理とデータタイプ（フィールドまたは変数）は以下の通りです。

最小値（数字）

最大値（数字）

強制値（英数字）

作成されたサイズの小さいJavaScriptは、ユーザのデータ入力の受け入れを阻害することなく（これはまだ4Dの責任です）「警告」ダイアログボックスを表示します。万がブラウザ上のデータ入力エリアに誤った値が含まれたまま、ユーザがボタン（OKやCancelなど）をクリックすると、ブラウザに警告メッセージが表示されます。



警告内容が解消され、ボタンをもう一度クリックをすると、ボタンの動作が受け入れられます。

完全なデータ入力管理が、（以前のバージョンのように）ユーザおよびカスタムモードのWebサーバ上で行われます。

JavaScriptデータ入力管理を使用するには、

- 1 「データベースプロパティ」ダイアログボックスで、「WebサーバII」タブをクリックする。
- 2 「Javascriptを入力制御に使用する」オプションを選択する。
デフォルトでは、このオプションは選択されていません。
- 3 「OK」ボタンをクリックする。

4D変数をスタティックホームページに挿入する新しいモード

データベースプロパティの「WebサーバII」のページにある「ブラケットの代りに4DVARコメントを使用する」オプションを使用すると、4D変数をスタティックホームページに挿入する時の記述方法を定義できます。

このオプションが指定されると、使用するべき構文は標準のHTML記述(<!--4DVAR MAVAR-->)になります。¹

このオプションが指定されない（デフォルト）と、使用するべき構文は、固有の方法である角カッコ付の記述([MAVAR])になります。

コンテキストを参照する新しいモード

デフォルトでは、4D Webサーバはブラウザからのリクエストごとに、現在のコンテキスト番号をブラウザに送ります。例えば、1つのページに2つの段落と1つのピクチャがある場合、4Dはコンテキスト番号を3回送ります。

バージョン6.5では、この動作を変更することが可能になり、ページを送る速度を上げることも可能になりました。新しいコンテキストモードでは、コンテキスト番号はドキュメントの基本URLに設定され、各オブジェクトごとに繰り返されません。

1. スペースを4DVAR文字と変数名の間に挿入しなければなりません。

新しいコンテキストモードを役立てるには、

- 1 データベースプロパティの「WebサーバI」ページで、「新しいコンテキスト参照モードを使用」オプションを選択する。
デフォルトでは、このオプションは選択されません。
- 2 「OK」ボタンをクリックして、変更を有効にするためにデータベースを一旦終了し、再度立ち上げる。

拡張 ASCII コードの直接送信

4D Webサーバは、デフォルトでダイナミックおよびスタティックホームページ内の拡張 ASCII コードをHTMLの基準に基づいて変換して送り、ブラウザによって解釈されます。

拡張 ASCII コードをWebサーバが変換せずに「そのまま」送るように設定することができます。このオプションは、英語以外のほとんどのオペレーティングシステム（特に日本語のシステム）で速度向上が図れます。

4D Webサーバに拡張 ASCII コードを直接送信させるには、

- 1 「データベースプロパティ」ダイアログで、「WebサーバII」タブをクリックする。
- 2 「拡張文字を直接送信」オプションを選択する。
- 3 「OK」ボタンをクリックする。

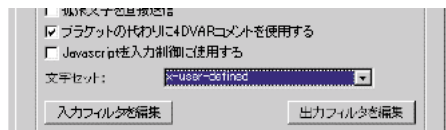
4D 内の変換文字セットの変更

バージョン6.5は、4Dにデータを入力または出力する際のASCII変換テーブル（Webフィルタ）を直接変更できます。以前のバージョンではCustomizer Plusで行っていたものです。

変換テーブルの変更は、「データベースプロパティ」ダイアログで「x-user-defined」文字セットが選択された場合にのみ可能です。

ASCII変換テーブルを変更するには、

- 1 「データベースプロパティ」ダイアログで「WebサーバII」タブをクリックする。
- 2 「文字セット」スクロールダウンリストで、「x-user-defined」オプションを選択する。
入力フィルタを編集および出力フィルタを編集ボタンが有効になります。



- 3 変更したいフィルタに対応するボタンをクリックする。

入力フィルタは、ブラウザによって4D Webサーバに送られた文字を変換します。出力フィルタは、4D Webサーバによってブラウザに送られる文字を変換します。次のダイアログボックスが現れます。



注：このダイアログボックスは、4Dのデータ読み込み/書き出しフィルタのダイアログボックスに似ていますが、これらは独立したものです。Web フィルタは、データ読み込み/書き出しフィルタには影響しません。

- 4 スクロールエリアで、フィルタにかけたいものを選択し、「読み込み...」をクリックするまたは、
以前に保存したWebフィルタを、「Load」ボタンをクリックすることによって読み込む（この場合ステップ8へ）。

- 5 「ASCIIコード」入力エリアで、文字の新しいASCIIコードを入力する。

注：データベースの互換性を確実にするために、フィルタのデフォルト値は4Dバージョン6.0.6で使用されたものと同じにされています（Customizer Plusで定義されていれば、「MapC」から）。

- 6 フィルタにかけたいすべての文字について、この操作を繰り返す。

- 7 必要であれば、フィルタをファイルに保存するために「保存...」ボタンをクリックする。

- 8 テーブル更新ボタンをクリックする。

- 9 「データベースプロパティ」ダイアログの「OK」ボタンをクリックする。
変更された、入力または出力のWebフィルタはこれで有効になりました。

スタティックホームページ用キャッシュ

4D Webサーバには、スタティックホームページ、GIF イメージ、JPEG イメージ（100KB 未満）およびスタイルシート（.css ファイル）が要求されれば、メモリからロードするキャッシュがあります。

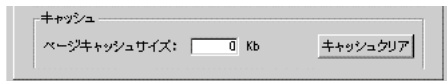
キャッシュを使用すると、スタティックホームページを送る際のWebサーバの能力が非常に増大します。

キャッシュはすべてのWebプロセスで共有されます。キャッシュのサイズはデータベースプロパティ内で設定できます。

デフォルトでは、スタティックホームページのキャッシュは有効になっていません（サイズは0になっています）。

スタティックホームページのキャッシュを有効にするには、

- 1 「データベースプロパティ」ダイアログの「WebサーバII」ページで、キャッシュのページキャッシュサイズ (KB単位) を設定する。



設定する値は、Webサイトのスタティックホームページの数とサイズならびにホストマシンが任意で持つリソースにより決定してください。

注：Webデータベースを使用している間、「Web CACHE STATISTICS」に記されているスタティックルーチンを使って、キャッシュの性能をチェックすることができます。例えば、キャッシュの使用率が100%に近いことに気づいた場合、これに割り当てられているサイズを増やすことを考えるかもしれません。/4DSTATSおよび/4DHTMLSTATSのURLもキャッシュの状態に関する情報を得ることができます。詳細は、この章の「Webサーバ管理URL」を参照してください。

- 2 「OK」ボタンをクリックする。

一旦キャッシュが有効になると、4D Webサーバはブラウザから要求されたページをキャッシュ内で探します。ページを見つけると即座に送り、見つからなければディスクからページをロードしてブラウザに送り、キャッシュ内にも置きます。

キャッシュがいっぱいで、追加のスペースが必要とされる場合は、最も需要の少ないものの中から最も古いページを最初に「アンロード」します。

キャッシュのクリア

いつでも、ページおよびイメージのキャッシュをクリアすることができます（例えば、スタティックホームページを変更して、それをキャッシュ内に再ロードしたい場合）。

これを行うには、データベースプロパティの「WebサーバII」ページ内の「キャッシュクリア」ボタンをクリックします。キャッシュは即座にクリアされます。

HTTP リクエスト用 IP アドレスの定義

データベースプロパティの「WebサーバI」ページで、WebサーバがHTTPリクエストについて返答するIPアドレスを定義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ようにすべてのIPアドレスに対し返答します(IPアドレスを指定しないオプション)。

IPを指定するオプションを選択すると、「IPアドレス」エリアが入力可となり、「194.166.100.101」のような特定のアドレスを入力できるようになります。この場合、サーバはこのアドレスから来たリクエストにのみ返答します。

この機能は、複数のTCP/IPアドレスを持つマシン上で動作している4D Webサーバのためのものです。これはほとんどのインターネットホストプロバイダの一般的なケースです。

Web サイトに関する情報

バージョン6.5では、4D Web サイトに関する情報を得ることができます。

特定の URL(/4DSTATS、/4DHTMLSTATS および/4DCASHECLEAR)を使ってサイトの管理ができます。

すべてのリクエストのログを作成できます。

Web サーバ管理 URL

バージョン6.5の4D Web サーバは、3つの特定の URL(/4DSTATS、/4DHTMLSTATS および/4DCASHECLEAR)を受け入れます。

注：これらの URL は、4D パスワードシステムが設定されている場合は、デザイナおよび管理者のみ使用できます。4D パスワードシステムが設定されていない場合は、すべてのユーザが使用可能になります。

/4DSTATS

/4DSTATSのURLは、純粋なテキストフォームで下記の情報を返します。

ヒット数（低レベル接続）

作成されたコンテキストの数

作成されなかったコンテキストの数（同時Web接続数がWebライセンス数に達した場合）

使用可能なWebライセンスの数

必要とされる追加ライセンスの数の推定¹

パスワードエラーとなった数

キャッシュ内に保存されているページの数

キャッシュの使用率(%)

スタティックホームページのキャッシュ内に、保存されてるページおよびGIF ファイルのリスト²

この情報は、サーバ機能をチェックし、段階的に各パラメータを最適化することを可能にします。

注：新しいコマンドである **WEB CACHE STATISTICS** は、キャッシュがスタティックホームページにどのように使用されているのか、に関する情報を得られるようにします（詳細は、第5章「ランゲージ」の「**WEB CACHE STATISTICS**」を参照してください）。

1. 理論上の最大値の推定（統計データから）

2. スタティックホームページおよびピクチャに関する詳細は、この章の「スタティックホームページ」を参照してください。

/4DHTMLSTATS

/4DHTMLSTATSのURLも、/4DSTATSのURLと同じ情報を純粋なテキストフォームで返します。違いは、最後のフィールドでキャッシュ内に存在するHTMLページのリストだけが返されることです（キャッシュされているGIFファイルのリストは含まない）。

/4DCASHECLEAR

/4DCASHECLEARのURLは、スタティックホームページとイメージのキャッシュを即座にクリアします。したがって、変更されたページを「強制的」に更新することができます。

接続ログファイル

バージョン6.5では、リクエストのログを取ることができます。ログは、ストラクチャファイルと同じ階層に、「weblog.txt」ファイルとして自動的に作成されます。このファイルは、ほとんどのWebサイト分析ツールで認識できる、CLF(Common Log File)フォーマットまたはNCSAフォーマットになります。

ファイルの各行は、次のようなリクエストを表わします。

```
host rfc931 user [DD/MMM/YYYY:HH:MM:SS] "query" state length
```

各フィールドはスペースで分離され、各行はCR/LF（文字コード13/文字コード10）で終わります。

host：クライアントのIPアドレス（例192.100.100.10）

rfc931：4Dでは生成しない情報で、常に「-（マイナス記号）」です。

user：認証されているユーザ名または「-（マイナス記号）」。

ユーザ名にスペースがあると「_（下線）」に置き換えられます。

DD：日付、MMM：月の名前の3文字の略号（Jan、Feb、...）YYYY：年、HH：時間、MM：分、SS：秒

日付と時間はサーバマシン上の値です。

query：クライアントから来たリクエスト（例えば、GET/index.htmHTTP/1.0）

state：サーバからの返答

length：返答データのサイズ（HTTPヘッダを除く）または0

注：性能上の理由から、ディスクに書き込まれる前にサイズ1KBのパケットとしてメモリ上に保存され、5秒間リクエストが発生しなければディスクに書き込まれます。

stateとして取り得る値は下記の通りです。

| | |
|------|-------------------------|
| 200: | OK |
| 204: | No contents |
| 302: | Redirection |
| 400: | Incorrect query |
| 401: | Authentication required |
| 404: | Not found |
| 500: | Internal error |

ログで作成される例：

192.100.100.10 - - [25/Jan/1998:12:54:06] "GET/index.htm" 200 6524

アドレスが192.100.100.10のWebクライアントが認証されませんでした。ページ
"index.htm"が要求され、送信しました（6,524バイト）。

192.100.101.25 - - [25/Jan/1998:12:54:09] "GET/123456.htm" 404 125

アドレスが192.100.101.25のWebクライアントが認証されませんでした。ページ
"123456.htm"を要求されたが見つけれませんでした（4Dは125バイトのメッセージ
を送りました）。

192.100.101.31 - - [25/Jan/1998:12:54:10] "GET/secret.htm" 401 0

アドレスが192.100.101.31のWebクライアントが認証されませんでした。ページ
"secret.htm"を要求され、サーバは認証要求をしました。

192.100.101.31 - ZZZZ [25/Jan/1998:12:54:11] "GET/secret.htm" 401 0

アドレスが192.100.101.31のWebクライアントが"ZZZZ"として認証されました。ページ
"secret.htm"を要求され、ユーザ名が不明です。

192.100.101.31 - ACI [25/Jan/1998:12:54:12] "GET/secret.htm" 200 2543

アドレスが192.100.101.31のWebクライアントが"ACI"として認証されました。ページ
"secret.htm"を要求され、送信しました（2,543バイト）。

警告：ログファイルはスプレッドシートまたは直接4Dへ読み込み可能です。しかし、デ
ータを読み込む前に、必ずWebサーバを停止する必要があります。

デフォルトではログファイルは作成されません。

すべてのWebリクエストのログファイルの作成を要求するには、

- 1 データベースプロパティの「WebサーバII」ページで、「ログを必要とする」オプションを選択する。
- 2 「OK」ボタンをクリックする。

HTML サポート

バージョン6.5のWebサーバでは、データベースオブジェクトのHTMLへの自動変換などのHTMLサポート機能が改善され、最適化されました。

この節では、これらの変更についての説明を行ないます。

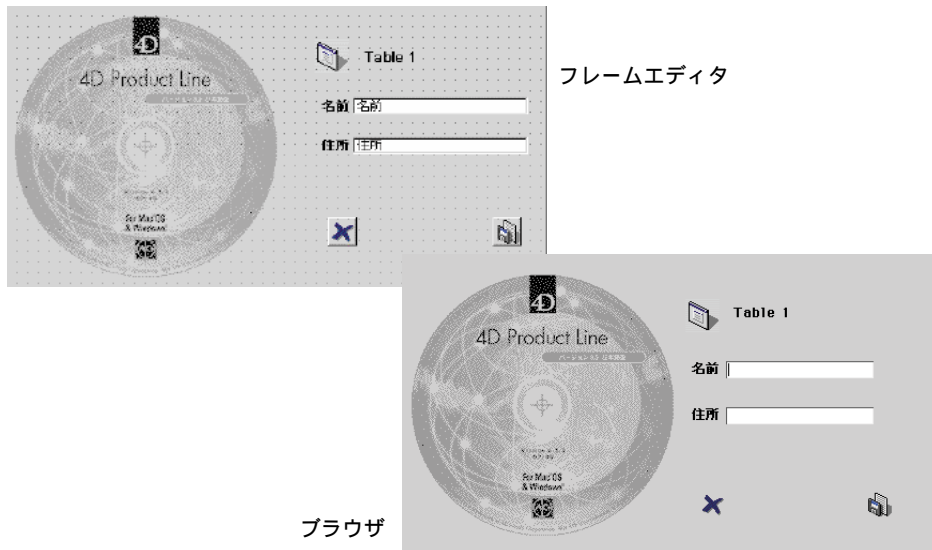
スタティックホームページ

4DはHTML 3.2と完全な互換性があります。

4DはスタティックホームページでのHTMLフレームをサポートします。

ダイナミックページ

ブラウザに送られる「フォームのオブジェクト配置」が改善されました。ブラウザ上のオブジェクト位置は4Dフォームと同じ位置を保ちます。



プラグインエリアは、最初にイメージまたはイメージマップに変換することによって、Web上でも表現可能になりました。イメージマップへの変換は、プラグインエリア内のマウスクリックの管理を可能にします（例えば、プラグイン4D Chartはイメージマップとして表現され、ACI_Pack External Clockエリアはイメージとして表現されます）。

注：4DフォームのプラグインエリアのWeb上での表現方法は、プラグインエディタの仕様によります。

フォームの0ページ上のオブジェクトを、ブラウザ上で見られるようになりました。

4Dピクチャ変数がHTMLページで参照されても、ブラウザ上で正確に表示されるようになります。

背景ピクチャ：4Dのフォームエディタ内の座標(0、0、x、x、)に挿入されたリプリーケートピクチャ（表示タイプが「繰り返し」）は、ブラウザの背景ピクチャとして送られます。

フォームエディタ

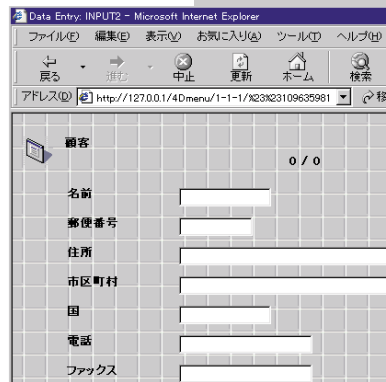
挿入されたピクチャ

座標

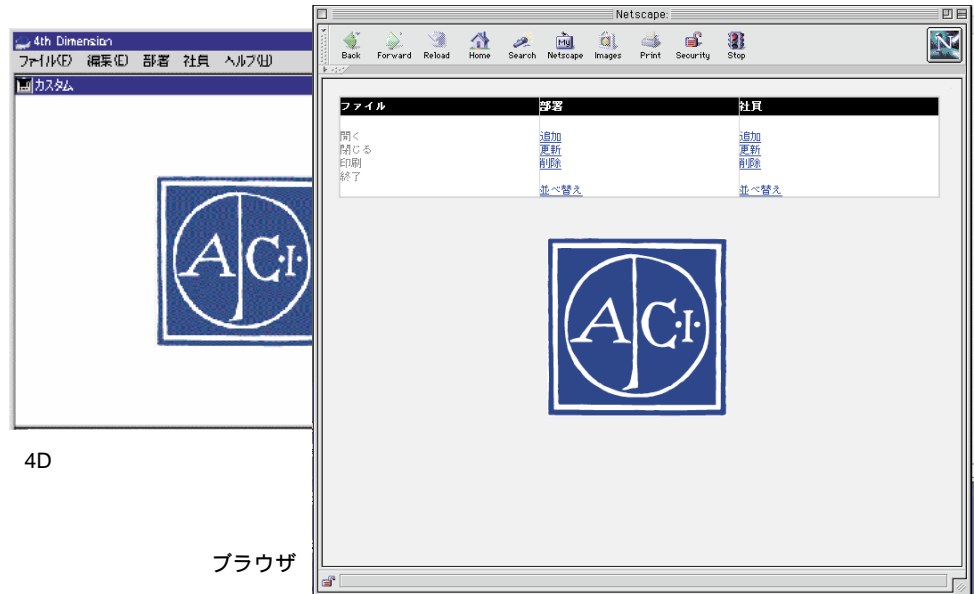
表示



ブラウザ



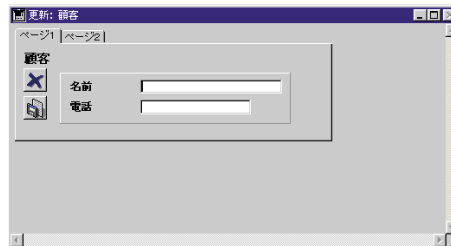
4Dメニューバーが表示されている時は、メニューバーに関連したピクチャがブラウザのメニューの下に現れます。



濃いピクチャは避ける必要があります。

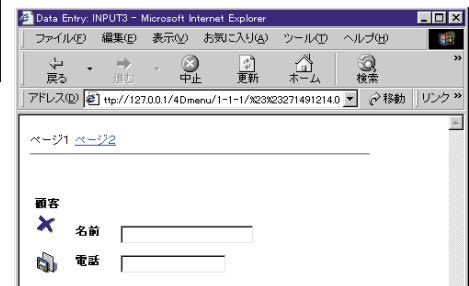
フォームエディタ内でピクチャとして表示されるボタンに関連したTipsが、ブラウザ上でも現れます（ブラウザがTipsの表示を許した場合）。

タブコントロール（配列またはオブジェクトプロパティで定義された値を使って作成されたもの）は、URL リストに変換されます。



4D

ブラウザ



配列要素が空白の場合、4Dはブラウザ上に、1、2、3...を表示します。

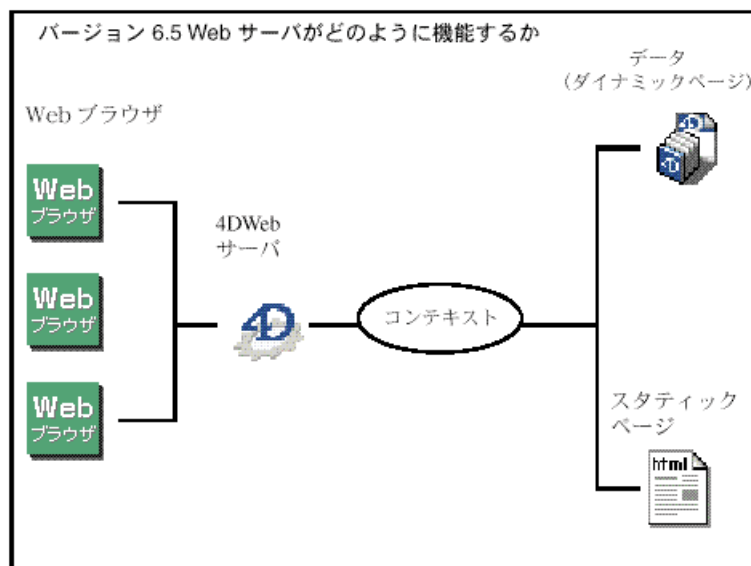
非コンテキストモード

バージョン 6.5 の Web サーバは、非コンテキストモードでも使用できるようになりました。このモードでは、昔ながらの HTTP サーバとなります。

コンテキストモードと非コンテキストモード

以前のバージョンは、Web プログラムの特長としてコンテキストの概念を用いていました。つまり、Web ブラウザがデータベースに接続すると、現在のセレクションや変数などを置いたコンテキストを作成します。ある意味では、各ブラウザは 4D クライアントとして扱われます（下図を参照してください）。

この方法によって、Web サーバはブラウザの動作を完全に管理し、データの信頼性を保証します。4D がスタティック HTML ページを送る時には、4D は昔ながらの HTTP サーバとして使用されてるため、あまり適切な方法ではありませんでした。この場合、ブラウザからのリクエストで作成されたコンテキストは、設定されているタイムアウトになるまで、接続しているものとして残ります。例えば、ブラウザが途中でサイトを抜けるとコンテキストや Web ライセンスは無駄になり、標準的なブラウザの機能（「再読み込み」や「戻る」など）が使用できませんでした。



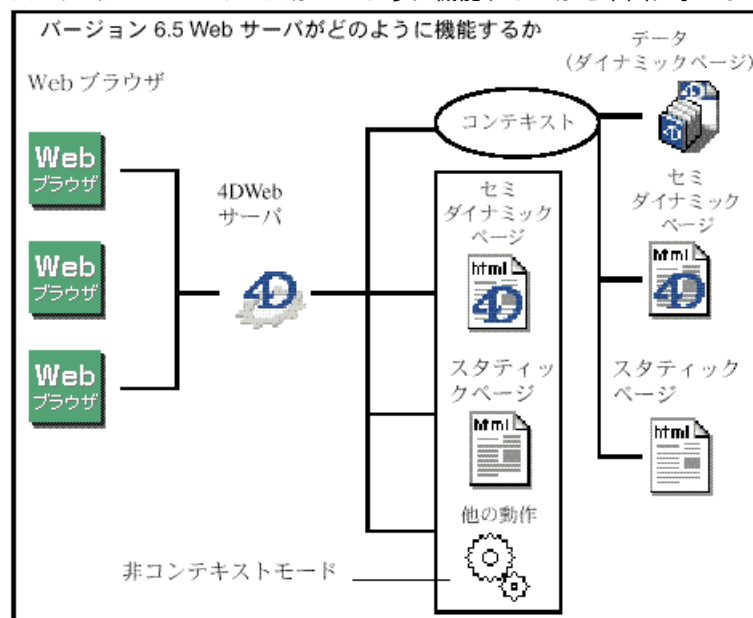
スタティックホームページを送る際に、4D Web サーバを昔ながらの HTTP サーバとして使用するには非コンテキストモードを採用します。このモードでの 4D Web サーバは、標準 HTTP サーバになります。すなわち、スタティック Web ページはコンテキストの保持や Web ライセンス¹を使用することなしに送られます。

1. Web プロセスが、PROCESS PROPERTIES コマンドを使用することによって Web ライセンスを消費するかどうかを知ることができます（詳細は、この章の「使用されているライセンスの数」を参照してください）。

スタティックホームページはキャッシュの恩恵を受けます（詳細は、この章の「スタティックホームページ」を参照してください）。この場合もまた、後述の **WEB CACHE STATISTICS** コマンドを使用して、これらのページの参照についての統計を得ることができます。

もちろん、記録に関するブラウザのナビゲーションを管理するために、コンテキストは常に使用されます。実際、4D Webサーバはコンテキストモード、非コンテキストモードまたは必要に応じてこれら2つのモード間での切り替えなど、希望するモードで 사용할ことができます（詳細は、この章の「非コンテキストモードからコンテキストモードへの切り替え」を参照してください）。

バージョン 6.5 Webサーバがどのように機能するかを下図に示しました。



注：セミダイナミックページは、Webサーバと4Dとの間で新しいタイプの通信を可能にしました。詳細は、この章の「セミダイナミックページ」を参照してください。

あるモードから別のモードへの切り替え

コンテキスト管理に関する限り、デフォルトでは、バージョン 6.5 Webサーバは以前のバージョンと同じように機能し、コンテキストがそれぞれのWeb接続で作成されます。

非コンテキストモードへの切り替えを、明確に指示する必要があります。

コンテキストモードから非コンテキストモードへの切り替え

データベースを使用している間、いつでもコンテキストモードから非コンテキストモードへ、またはその逆の切り替えをすることができます。

注：デフォルトでは、4D Web サーバはコンテキストモードで開始しますが、非コンテキストモードでの開始を定義することも可能です。詳細は、この章の「スタートアップ時に非コンテキストモードを定義する」を参照してください。

4D Web サーバを非コンテキストモードへ切り替えるには、3つの解決法があります。

新しいコマンド **SEND HTML BLOB** を実行し、オプションパラメータに **True**(noContext) を渡します。例えば、

SEND HTML BLOB (MyBlob; ".HTM" ;True)

4D は、BLOB を非コンテキストモードで送り、コンテキストを管理するプロセスはすぐに停止されライセンスは開放されます。

注：最後のパラメータを渡さないことは、False を渡すことと同じです。既に非コンテキストモードの場合、パラメータは無視されます。現在のモードは、**Web Context** を使用して調べることができます（詳細は、第5章「ランゲージ」の「**Web Context**」関数の説明を参照してください）。

新しいコマンド **SEND HTTP REDIRECT** を実行します。

このコマンドは、パラメータに渡された URL へのリクエストをリダイレクトします。コンテキストモードから呼び出されると、Web プロセスを閉じてライセンスを開放します。

/4DACTION で始まる URL を呼び出します（詳細は、この章の「4DACTION タグ」を参照してください）。

非コンテキストモードは、コンテキストモードを呼び出す URL が何もない時に使用することができます。

非コンテキストモードからコンテキストモードへの切り替え

一般的に、4D のコンテキストモードは Web サーバがダイナミックページを作成して送らなければならないと、すぐにデータベースに要請します。

非コンテキストモードからの 4D データベースの呼び出しは、4D 6.0.x でのように、特定の URL: "/4DMETHOD/MyMethod" を通して行われます。この URL をスタティックホームページ（例えばボタン内）に置きます。

この URL が有効になると、4D Web サーバは新しいコンテキストを作成して下記のことを実行します。

On Web Authentication データベースメソッドが存在している場合、実行されます。

On Web Connection データベースメソッドが存在している場合、実行されます¹。

メソッドが、新しく作成されたコンテキスト内で実行されます。

Web サーバは、これまでの 4D のバージョンと同様に機能します。コンテキストモードは、非コンテキストモードを有効にするコマンドが呼び出されない限り、そのモードで動作します。

1. この特別なケースでは、\$1 は "/"（スラッシュ）ではなく、"/4DMETHOD/MyMethod" になります。

注：バージョン6.5では、On Web Connectionデータベースメソッドが、4D内で「エントリーポイント」（非コンテキストモードからの）の役割も果たします。データベースメソッド内の呼び出し元に応じて、異なった指示を与えることができます（詳細は、この章の「On Web Connectionデータベースメソッドの変更」を参照してください）。

使用されているライセンスの数

実行されている動作に応じて、特定のWebプロセスはWebライセンスを使用します。

ライセンスは4Dによって内部的に管理されます。しかし、コンテキストモード、非コンテキストモードに関係なく、どのWebプロセスでも**PROCESS PROPERTIES**コマンドを使うことによって、使用されているライセンスの数を知ることができます。このコマンドは、新しいパラメータoriginに、Webライセンスが使用されているか（-11:Web Process with license）または使用されていないか（-3:Web Process with no license）を返します。

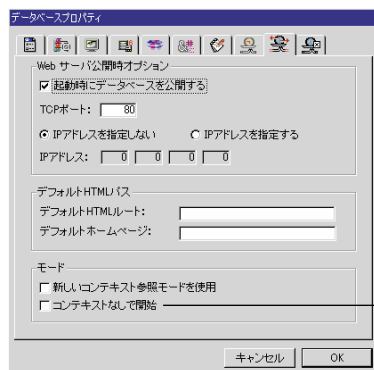
スタートアップ時に非コンテキストモードを定義する

4D Webサーバを、自動的に非コンテキストモードで開始することができます。これは、ユーザがデータベースに接続する時にコンテキストが作成されないことを意味します。

スタートアップ時に非コンテキストモードを定義するには、

- 1 「データベースプロパティ」ダイアログで、「WebサーバI」タブをクリックする。
- 2 「コンテキストなしで開始」オプションを選択する。

デフォルトでは、このオプションは選択されておらず、4D Webサーバはコンテキストモードでスタートします。



「非コンテキストモード」で始するオプション

- 3 ダイアログボックスを受け入れる。

4D Webサーバは、非コンテキストモードでホームページを送ります。

データベースプロパティの「デフォルトホームページ」エリアで、カスタムホームページを選択すると（「デフォルトホームページの定義」を参照）、そのホームページが送られます¹。

-
1. ブラウザから指定されたURLが、/（スラッシュ）で終わる場合、4Dはホームページの名前を追加して送ろうとします。例えば、URLが「/Folder/」であり、デフォルトホームページが「MyPage.HTM」の場合、4Dは、「/Folder/MyPage.HTM」を探します。これが見つからない場合は、On Web Connectionデータベースメソッドが呼び出されます。
-

デフォルトホームページを定義していなかったり、指定されたホームページが存在しない場合は、On Web Connection データベースメソッドが（非コンテキストモードで）呼び出されます。また、各 Web ユーザ固有の接続方法を管理することもできます。例えば、データベースに接続する各ユーザ用のホームページを指定するために、新しいコマンド **SET HOME PAGE** を使用することもできます。

セミダイナミックページ

バージョン 6.5 は、セミダイナミック方式で作成される新しいタイプの Web ページを採用しました。セミダイナミックページは、その内容の全部または一部が 4D によって実行されたアクションから生成される Web ページです。

セミダイナミックページは、コンテキストモードまたは非コンテキストモードのどちらでも使用できます。しかし、実際にはほとんどが非コンテキストモードで使用されます。このモードで 4D はブラウザのナビゲーションに関連した、どんな変数やセレクションなども保持しません。セミダイナミックページの目的は、4D の利点と非コンテキストモードに関連した利点を Web ページに取り入れることです。

セミダイナミックページは、下記の 3 つの方法で作成することができます。

/4DCGI/<action> という URL を 4D Web サーバに送る。

HTML ページの拡張子に「.shtm」または「.shtml」を付けて送る。

「/4DACTION/」タグを使用する（この特定のタグについては、この章の「4DACTION タグ」を参照してください）。

URL / 4DCGI / <action>

4D Web サーバが 4DCGI / <action> の URL を受け取ると、この URL をそのまま \$1 にセットし On Web Connection データベースメソッドを呼び出します。

4DCGI / の URL は、どのファイルにも対応しません。この役目は 4D を呼び出すことです。<action> のパラメータは、どのタイプの情報も持つことができます。

URL は、どのような動作でも実行させることができます。On Web Connection データベースメソッドまたはそのサブメソッド内で \$1 の値をテストし、4D に適切な動作をさせるだけです。例えば、レコードの作成、検索、ソートまたは GIF イメージを作成し、完全にカスタム HTML ページを構築することができます。この URL をどのように使用するか例は、**PICT TO GIF** および **SEND HTTP REDIRECT** コマンドの説明を参照してください。

データを送るコマンド（**SEND HTML FILE**、**SEND HTML BLOB** など）を使用して、問い合わせの結果を返されなければなりません。

警告：ブラウザを停止させないように、可能な限り短い動作で実行させることを心掛けてください。

「.shtm」ファイル

拡張子が「.shtm」（または「.shtml」）であるHTMLページを、4D Web サーバがブラウザに送る前に、4D変数への参照とページ内に存在するかもしれない4DACTIONのコメントを解析します。

これは、HTMLページの作成に4D内での処理実行を可能にし、スタティックホームページ内のダイナミックリファレンスの更新を確実にします。しかし、ブラウザが停止しないように、可能な限り短い動作で実行させることを心掛けてください。

注：非コンテキストモードでは、4Dピクチャ変数への参照を行なうことはできません。

WEB CACHE STATISTICS コマンドの例を参照してください。

4DACTION タグ

4DACTION タグは、スタティックHTMLページを送る際に4Dメソッドの実行を可能にします。このタグは3通りの使用方法があります。

HTMLコメントとして使用する(<!--4DACTION...-->)

URLとして使用する()

HTMLフォームの送信（フォームアクション）

HTMLコメントとしての4DACTION

スタティックホームページ内のHTMLコメントとして/4DACTION/MyMethod/MyParam タグが存在するということは、MyParamパラメータを文字列で\$1にセットしMyMethodメソッドの実行を要求するということです。ホームページをロードする際に、4Dは、On Web Authentication データベースメソッドがあれば呼び出し、関数値としてTrueが返った場合、4Dは指定されたメソッドを実行します。

メソッドは\$0にテキストを返します。返された文字列がASCIIコードの1で始まる場合、4DにHTML文として考慮されます（同じ原理は変数にも当てはまります）。

警告：「ブラケットの代わりに4DVARコメントで使用する」オプションが、データベースプロパティで選択されていなければなりません（詳細は、この章の「4D変数をスタティックホームページに挿入する新しいモード」を参照してください）。

ページ内容の解析は、**SEND HTML FILE**(.htm、.html、.shtm、.shtml)または**SEND HTML BLOB**(タイプtext/htmlのBLOB)が呼び出された時にも行われます。

非コンテキストモードで、URLが「.shtm」または「.shtml」のどちらかの拡張子を持つファイル（例えば、http://www.server.com/dir/page.shtm）を指す場合にも解析が行なわれます。

注：コンテキストモードでは、メソッドはコンテキスト内で実行されます。

例えば、スタティックホームページに"Today is <!--4DACTION/MYMETHOD/ MYPARAM-->" コメントを挿入したとします。

4D はページをロードする時に、On Web Authentication メソッドが存在していれば呼び出し後、文字列 "/MYPARAM " を \$1 の値として MYMETHOD メソッドを呼び出します。

メソッドは \$0 にテキスト (例えば、"98/10/28") を返すため、"Today is <!--4DACTION/MYMETHOD/MYPARAM-->" の式は、"Today is 98/10/28" になります。

C_TEXT(\$0) このパラメータは常に宣言する必要あり

C_TEXT(\$1) このパラメータは常に宣言する必要あり

\$0=String(Current date)

注：4DACTION で呼び出されたメソッドは、インタフェースエレメント (DIALOG、ALERT...) を呼び出してはいけません。

どのモードで実行していようと、現れた順に 4D が実行するメソッド内で、更にドキュメント内で参照される多くの変数の値を設定するメソッドを呼び出すことができます。

注：スタティックホームページには、好きなだけ多くの <!--4DACTION....--> コメントを挿入することができます。

URL としての 4DACTION

4D が /4DACTION/MYMETHOD/MYPARAM リクエストを受け取ると、On Web Authentication メソッドデータベースがあれば呼び出し、関数値として True が返った場合、MYPARAM パラメータを文字列で \$1 にセットし MYMETHOD メソッドが実行されます。URL の構文は、下記の形でなければなりません。

```
<A HREF="/4DACTION/MYNAME/METHOD/PARAMS">DoSomething</A>
```

注：これは、非コンテキストモードに適用されているということ以外は、/4DMETHOD と同一のもので (詳細は、『4th Dimension ランゲージリファレンス』の「Web サーバ コマンド」の章を参照してください)。

コンテキストモードでは、4D が /4DACTION タイプのリクエストを受け取るとコンテキストはアボートされ、メソッドはコンテキスト外で実行されます。これは、非コンテキストモードで /4DMETHOD を呼び出すのとは逆のもので (スタティックに関する説明を参照してください)。

注：/4DACTION で呼び出されたメソッドは、インタフェースエレメント (DIALOG、ALERT...) を呼び出してはいけません。

下記の指示をHTMLページに挿入します。

```
<IMG SRC="/4DACTION/PICTFROMLIB/1000">  
PICTFROMLIB メソッドは下記のようになります。  
C_TEXT `このパラメータは常に宣言されなければなりません  
C_PICTURE($PictVar)  
C_BLOB($BlobVar)  
C_LONGINT($Number) ` ` $1からピクチャの番号を取り出します  
$Number:=Num(Substring($1;2;99))  
GET PICTURE FROM LIBRARY($Number;$PictVar)  
PICT TO GIF($PictVar;$BlobVar)  
SEND HTML BLOB($PictVar;"Pict/gif")
```

4DACTION を POST する Form

非コンテキストモードで、データをWebサーバへ送るためにPOSTするFormを使用したい場合、しばしば、付加価値としての機能を提供します。POST¹タイプはこれらに関連付けられなければならない、Formの動作は、/4DACTION/MethodNameで始まるのが絶対条件です。

この場合、WebサーバがPOSTされたFormを受け取る時、COMPILER_WEBプロジェクトメソッド（これが存在する場合、下記参照）を、次にはOn Web Authenticationデータベースメソッド（存在する場合）を呼び出し、Trueを返した場合に、MethodNameメソッドが実行されます。4Dは、Form内に存在するHTMLフィールドを解析し、これらの値を検索し、自動的に4D変数をこれらの内容で置き換えます。Form内のフィールドおよび4Dの変数は同じ名前であればなりません。

注：/4DACTIONで呼び出されたメソッドは、インタフェースエレメント（DIALOG、ALERT...）を呼び出してはいけません。

フォーム内で適用するHTML構文は、下記の形になります。

```
Form内の動作を定義するには、  
<FORM ACTION="/4DACTION/MethodName"METHOD=POST>  
  
Form内のフィールドを定義するには、  
<INPUT TYPE=Field type NAME=Field name VALUE="Default value">  
4DはForm内の各フィールドの値を同じ名前の変数に設定します。
```

Formのオプション（例えばチェックボックス）には、関連する変数を、1が選択されていれば1に、そうでなければ0に設定します。

数値データの入力では、4Dはフィールドの値を文字列から数値に変換します。

1. 4Dは、GETタイプをPOSTするFormも受け入れます。しかし、この場合、作成されたURLのHTML文を手動で処理しなければなりません。さらに、COMPILER_WEBメソッドは呼び出されません。4Dは、マルチパート/フォームデータのFormも受け入れることに注意してください。

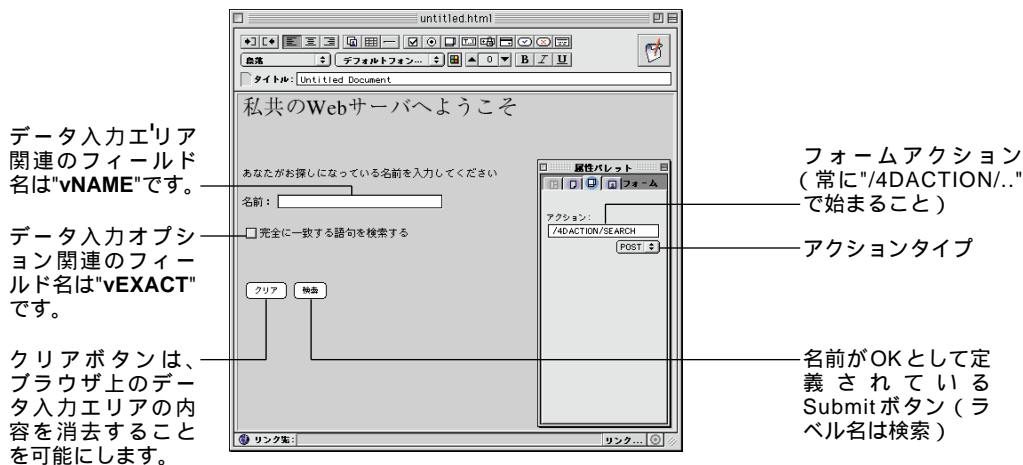
注：Form内のフィールドにOKと名付けられると（例えばSubmitボタン）、フィールドの値が空でなければ、システム変数OKが1に設定されます。そうでない場合は0が設定されます。

COMPILER_WEB メソッド：4D WebサーバがPostされたFormを受け取ると、COMPILER_WEBと呼ばれるプロジェクトメソッド（それが存在している場合）を呼び出します。このメソッドには、名前がForm内のフィールド名と同じ名前の全変数のタイプ設定や、必要であれば変数の初期化指定が含まれていなければなりません。これはデータベースをコンパイルする時に4D Compilerによって使用されます。

COMPILER_WEB メソッドはすべてのFormに共通です。

非コンテキストモードで開始されて使用されている4D Webデータベースがあり、スタティックHTMLページを使用してレコードの検索ができるとします。検索ページを"search.htm"とし、結果を返すページを"results.htm"とします。POSTタイプは、ACTIONとして/4DACTION/PROCESSFORMがページに記述されています。

下記は、Adobe® PageMill™のHTMLエディタ内で現れるページです。



下記は、このページに対応するHTMLコードです。

```
<FORM ACTION="/4DACTION/PROCESSFORM"METHOD=POST>
<INPUT TYPE=TEXT NAME=VNAME VALUE=""><BR>
<!--通常は解釈上の理由から VALUE にボタン名を入れますが、4Dで使用する場合は
VALUE に番号を入れなければなりません-->
<INPUT TYPE=CHECKBOX NAME=EXACT VALUE="1">Whole word<BR>
<!--OKは、特別な意味があります-->
<INPUT TYPE=SUBMIT NAME=OK VALUE="Search">
</FORM>
```

データ入力中に、データ入力エリア内に"ABCD"をタイプし、オプションをチェックして Search ボタンをクリックします。

4D は、次に下記のような COMPILER_WEB プロジェクトメソッドを呼び出します。

C_TEXT(VNAME)

VNAME:=""

C_LOGINT.(vEXACT)

vEXACT:=0

OK:=0 `特別なケース

この例では、VNAME には文字列"ABCD"が入り、vEXACT は1となり、OK は1になります (ボタンの名前が OK であるためです)。

4D は On Web Authentication データベースメソッド (存在する場合) を呼び出し、次に下記のように PROCESSFORM プロジェクトメソッドが呼び出されます。

If (OK=1)

If (vEXACT=0) ` オプションが選択されなかった場合

 vNAME:=VNAME+"@"

End if

QUERY([Jockeys];[Jockeys]Name=vNAME)

 vLIST:=**Char**(1) ` リストを HTML で返します

FIRST RECORD([Jockeys])

While (**Not**(**End selection**([Jockeys])))

 vLIST:=vLIST+[Jockeys]Name+" "+[Jockeys]Tel+"
"

NEXT RECORD([Jockeys])

End while

SEND HTML FILE("results.htm") ` リストを results.htm へ送ります

 :

End if

On Web Connection データベースメソッドの変更

On Web Connection データベースメソッドは変更され、6つの引数を受け取り、4D Web サーバ内での役割が強化されました。

新しいパラメータ

On Web Connection データベースメソッドは、既存のもの(\$1、\$2)の他に4つの新しい引数(\$3、\$4、\$5、\$6)も受け取ります。これらの引数は、On Web Authentication データベースメソッドによって受け取られるものと同一です (詳細は、この章の「On Web Authentication データベースメソッド」を参照してください)。

Text タイプであるこれらの引数は、接続に関してのいろいろな情報を提供します。

| 引数 | タイプ | 内容 |
|-----|------|-------------|
| \$1 | テキスト | URL |
| \$2 | テキスト | HTTPヘッダ |
| \$3 | テキスト | ブラウザのIPアドレス |
| \$4 | テキスト | サーバのIPアドレス |
| \$5 | テキスト | ユーザ名 |
| \$6 | テキスト | パスワード |

\$1と\$2の引数に関するより詳しい情報は、4Dのドキュメント内にある On Web Connection データベースメソッドの内容を参照してください。

\$3から\$6の引数に関する詳細は、スタティックに関する説明を参照してください。

注：ブラウザによって送られたユーザ名が4Dのユーザとして登録されている場合、引数\$6（ユーザのパスワード）へはセキュリティの関係上何も渡されません。

On Web Connection と非コンテキストモード

以前のバージョンのように、On Web Connection データベースメソッドは、ブラウザがデータベースに接続する時に呼び出されます。また、これはWebサーバの新しい「非コンテキスト」モード時の4Dへのエン트리ポイントとしての役目も果たしています。

実際に、非コンテキストモードからコンテキストモードへの切り替えは、On Web Connection データベースメソッドを通して行なわれます（詳細は、この章の「非コンテキストモード」を参照してください）

注：非コンテキストモードでは、インタフェースエレメント（ALERT、DIALOG...）を表示する4Dコマンドは処理を終了させます。

On Web Connection データベースメソッドは、下記の場合に呼び出されます。

ブラウザが4D Webサーバに接続する場合（4D 6.0.xと同様に）：データベースメソッドは、/<action>...のURLで呼び出されます。

非コンテキストモードからコンテキストモードへ切り替える場合：データベースメソッドは、/4DMETHOD/MethodNameのURLで呼び出されます。

4Dが/4DCGI/<action>のURLを通して、セミダイナミックページに呼び出される場合：データベースメソッドはURLで呼び出されます（詳細は、この章の「セミダイナミックページ」を参照してください）。

Webページが非コンテキストモードで呼び出され、<path>/<file>タイプのURLが見つからない場合：データベースメソッドはURLで呼び出されます。

Webページが非コンテキストモードで呼び出され、<file>/タイプのURL指定で、デフォルトのホームページが定義されていない場合：データベースメソッドはURLで呼び出されます。

バージョン 6.5 には、実行しなければならない処理を判断できるように、On Web Connection データベースメソッドがどこから呼び出されたのかを知るための新しい関数 **Web Context** が追加されました（詳細は、「**Web Context**」コマンドの説明を参照してください）。

結果として、バージョン 6.5 内の On Web Connection データベースメソッドを、下記のよう構成にすることを提案します。

```
`On Web connection データベースメソッド
If (Web Context)    `もし、コンテキストモードなら
    WithContext ($1;$2;$3;$4;$5;$6)
        `WithContext メソッドは、4D 6.0.x 内にあった
        `On Web connection データベースメソッドのすべてを含む
Else
    NoContext ($1;$2;$3;$4;$5;$6)
        `NoContext メソッドは、リクエスト（一般的には短いもの）
        `の非コンテキスト処理を実行する
End if
```

4th Dimensionおよび4D Serverのバージョン6.5は、多くの新しいコマンドを提供しています。既存のコマンドのいくつかも変更されています。

次の3つのフォームイベントが追加されました。

「On After Keystroke」、 「On Timer」 および 「On Resize」。

既存のフォームイベント「On Clicked」および「On Double Clicked」も変更されました。

バージョン6.5では4Dのデバッガも強化されています。

4Dのプログラミング言語に関連するすべての変更について説明します。

注：バージョン6.5では、（変数用としての）セマフォの名前には、接頭辞(\$、<>)を含めて30文字まで持つことができます。

リスト項目は、255文字まで持つことができるようになりました。

新しいコマンド

このマニュアル内では、テーマ順にコマンドの説明をします。

配列

BOOLEAN ARRAY FROM SET

BOOLEAN ARRAY FROM SET(ブール配列{;セット})

| 引数 | タイプ | 説明 |
|-------|-------|-------------------------|
| ブール配列 | ブール配列 | レコードがセット内にあるかどうかを示す配列 |
| セット | 文字列 | セットの名前 |
| | | この引数を省略した場合 UserSet を使用 |

説明

このコマンドは、テーブル内の各レコードが指定されたセットに含まれているかどうかのブール配列を作成します。ブール配列には、セットに含まれているレコード数より多くの要素数があります。配列内の要素は、レコードがテーブル内で作成された順番で並べられています。

配列の各要素は次の通りです。

対応するレコードがセットに属している場合は、True

対応するレコードがセットに属さない場合は、False

配列要素0はレコード番号0に対応し、配列要素1はレコード番号1に対応します。

引数セットを指定しない場合、コマンドは現在のプロセスの UserSet を使用します。

LONGINT ARRAY FROM SELECTION

LONGINT ARRAY FROM SELECTION({テーブル;}レコード番号配列{;セクション名})

| | | |
|----------|--------|--|
| 引数 | タイプ | 説明 |
| テーブル | テーブル | テーブル この引数を省略した場合は現在のデフォルトテーブル |
| レコード番号配列 | 倍長整数配列 | レコード番号の配列 |
| セクション名 | 文字列 | 命名セクション この引数を省略した場合は現在のカレントセクションの名前 |

説明

このコマンドは、セクション名に属している各レコードに対する（絶対）レコード番号の配列を、レコード番号配列として作成します。

セクション名を省略した場合、コマンドはテーブルのカレントセクションを使用します。この場合は最初の引数としてテーブルを指定しなければなりません。

注：配列要素0は-1に初期化されます。

フォームイベント

SET TIMER

SET TIMER(チック数)

| 引数 | タイプ | 説明 |
|------|------|----------|
| チック数 | 倍長整数 | 設定するチック数 |

説明

このコマンドは、現在のプロセスで、指定されたTickごとにOn Timer フォームイベントを発生させます。

注：この新しいフォームイベントについての詳細は、「On Timer」の説明を参照してください。

このコマンドがフォームを表示していないプロセス内で呼び出されても何も変化はありません。

4DのWebサーバで、このコマンドとOn Timer フォームイベントを使用し、4D フォームの再送処理をすることができます。

この方法は、バンド幅を節約しながら「リアルタイム」でHTMLページを更新することができます。この場合のフォームの更新は、実際には自動ではありません。**REDRAW** コマンドを呼び出す必要があります。そうすると、データが変更された場合にのみ**REDRAW**を呼び出すことによって、システムを最適化できます。

JavaScriptを解釈するブラウザは、ページを自動的に書き直すことが可能です。**SET TIMER**によって定義された時間がブラウザによって使用されます。これは数秒間（5が現実的な値です）であり、Webプロセスのタイムアウト時間でなければなりません。下記の2番目の例を参照してください。

On Timer フォームイベント発生を取り消すには、チック数に0をセットした**SET TIMER**をもう一度実行します。

例題

1. フォームが画面に表示されている時に、コンピュータが3秒毎に警告音を鳴らすようにしたいと仮定します。その場合、下記のようにフォームメソッドを書きます。

```
If (Form event=On Load)  
    SET TIMER(60*3)  
End if  
If (Form event=On Timer)  
    BEEP  
End if
```

- 2 Webサーバが、Webブラウザ上に表示されている4Dフォームを、5秒毎に更新するようにしたいと仮定します。その場合、下記のようにフォームメソッドを書きます。

```
If(Form event=On Load)  
    SET TIMER(60*5)  
End if  
If (Form event=On Timer)  
    ... `データが変更されている場合のみ次の行を実行させるために、  
    `ここにデータ変更判断文を置くことができます  
    REDRAW  
End if
```

参照

On Timer フォームイベント

Get edited text

Get edited text Text

| 引数 | タイプ | 説明 |
|----|-----|----|
|----|-----|----|

| | | |
|------------------|--|--|
| この関数には、引数はありません。 | | |
|------------------|--|--|

| | | |
|-----|------|-------------|
| 戻り値 | テキスト | 入力されているテキスト |
|-----|------|-------------|

説明

この関数は、フォームオブジェクトに入力されているテキストを返します。

この関数は、主に入力されているテキストを検査するために、新しいフォームイベント On After Keystroke と共に使用されます。また、On Before Keystroke フォームイベントと共に使用することもできます。

注：新しいフォームイベントである On After Keystroke に対応するために、従来のイベントである On Keystroke は On Before Keystroke となりました。フォームイベントの新しい特長に関する詳細は、この章の「フォームイベント」を参照してください。

フォームオブジェクト内でテキストを入力しないコンテキストでは空白を返します。

例題

1. 下記のメソッドは、入力される文字を自動的に大文字に変換します。

```
If (Form event=On After Keystroke)
  [Trips]Agencies:=Uppercase(Get edited text)
End if
```

2. テキストフィールドへの入力文字の処理例があります。これは、他のテキストフィールド（ここでは"Words"という名前）に、入力された文章の単語を取り出すというものです。これを行うには、フィールドのオブジェクトメソッド内に下記のコードを書き込みます。

```
If (Form event=On After Keystroke)
  $RealTimeEntry:=Get edited text
  PLATFORM PROPERTIES($platform)
  If ($platform#3) `Macintosh または Power Macintosh
    Repeat
      $DecomposedSentence:=Replace string($RealTimeEntry;Char(32);
                                           Char(13))
    Until (Position(" ";$DecomposedSentence)=0)
  Else `Windows
    Repeat
      $DecomposedSentence:=Replace string($RealTimeEntry;Char(32);
                                           Char(13)+Char(10))
    Until (Position(" ";$DecomposedSentence)=0)
```

End if

[Example]Words:=\$DecomposedSentence

End if

注：この例は、単語がスペース(Char (32))によって区切られていると仮定しているため完全なものではありません。完全な解決法としては、すべての単語を抽出するように他のフィルタを付加する必要があります（カンマ、セミコロン、アポストロフィーなどの区切り）。

データ読み込みとデータ書き出し

注：バージョン6.5のデータ読み込みおよびデータ書き出しの機能は変更されました。2つの新しいコマンドは、バージョン6.5の新しい多くの利点を提供します。

IMPORT DATA

IMPORT DATA(ファイル名{;プロジェクト{; *}})

| 引数 | タイプ | 説明 |
|--------|------|---|
| ファイル名 | 文字列 | 読み込みファイルへのパス |
| プロジェクト | BLOB | インポートプロジェクトの内容 インポートプロジェクトの新しい内容 (*引数が渡された場合) |
| * | | 「データ読み込み」ダイアログボックスを表示し、プロジェクトを更新します。 |

説明

このコマンドは、データを<ファイル名>ファイルから読み込めるようにします。4Dは次のフォーマットのデータを読み込むことができます：

テキスト、固定長のテキスト、SYLK、DIF、DBF(dBase)および4th Dimension

空白をファイル名に渡すと、**IMPORT DATA**は標準「ファイル保存」ダイアログボックスを表示して、ユーザが読み込むファイルの名前、タイプおよび位置を定義することができます。ダイアログボックスが受け入れられると、Documentシステム変数にファイルパスがセットされます。ユーザがキャンセルをクリックすると、コマンドの実行は停止されて、システム変数OKは0になります。

引数<プロジェクト>を省略した場合、「データ読み込み」ダイアログボックスが表示され、インポートパラメータを定義するか、または既存の定義ファイルからインポートプロジェクトを読み込むことができます。

注：インポートプロジェクトには、読み込むテーブルやフィールド、区切り符号（デリミッタ）のようなインポートに関するすべてのパラメータが含まれています。これらのパラメータは「データ読み込み」ダイアログボックス内で定義します。プロジェクトはディスクに保存され、読み込んで使用することができるようになります。「データ読み込み」ダイアログボックスについての詳細は、この章の「データ読み込みとデータ書き出し」を参照してください。

有効なインポートプロジェクトを持つBLOBをプロジェクト引数に渡した場合、ユーザの操作なしに直接データ読み込みが実行されます。プロジェクトは「データ読み込み」ダイアログボックス内で前もって定義し、保存しておかなければいけません。これを行うには、2つの方法があります。

プロジェクトをディスクに保存後、**DOCUMENT TO BLOB** コマンドを使用して、フィールドまたはBLOB変数にセットします。

空白のプロジェクト引数およびオプション引数*を指定した**IMPORT DATA** コマンドを実行し、プロジェクト引数のBLOBに保存します（下記参照）。この方法は、ディスク上からプロジェクトを読み込む必要はなく、データと共にプロジェクトを保存することができます。

オプションの引数*が指定されていれば、プロジェクト内に定義されたパラメータと共に、「データ読み込み」ダイアログボックスを表示します。これは、パラメータの1つまたはそれ以上を変更できる可能性を持ちながら、前もって定義されたプロジェクトを使用できるようにするものです。さらに、「データ読み込み」ダイアログボックスを閉じた後に、プロジェクト引数は、「新しい」プロジェクトのパラメータを持つことができ、新しいプロジェクトをBLOBフィールドやディスク上などに保存することができます。

データ読み込みが正常に終了すると、システム変数OKは1になります。

参照

EXPORT DATA

EXPORT DATA

EXPORT DATA(ファイル名{;プロジェクト{;*}})

| 引数 | タイプ | 説明 |
|--------|------|--|
| ファイル名 | 文字列 | 書き出しファイルへのパス |
| プロジェクト | BLOB | エクスポートプロジェクトの内容 |
| | | エクスポートプロジェクトの新しい内容 (*引数が渡された場合) |
| * | | 「データ書き出し」ダイアログボックスを 表示し、プロジェクトを更新します。 |

説明

このコマンドは、データを<ファイル名>ファイルへ書き出せるようにします。4Dは次のフォーマットで書き出すことができます：

テキスト、固定長のテキスト、SYLK、DIF、DBF(dBase)および4th Dimension

空白をファイル名に渡すと、**EXPORT DATA**は標準「ファイル保持」ダイアログボックスを表示して、ユーザが書き出すファイルの名前、タイプおよび位置を定義することができます。ダイアログボックスが受け入れられると、Documentシステム変数にファイルパスがセットされます。ユーザがキャンセルをクリックすると、コマンドの実行は停止されて、システム変数OKは0になります。

引数<ファイル名>を省略した場合、「データ書き出し」ダイアログボックスが表示され、エクスポートパラメータを定義するかまたは既存の定義ファイルからエクスポートプロジェクトをロードすることができます。

注：エクスポートパラメータには、書き出すテーブルやフィールド、区切り符号（デリミッタ）のようなエクスポートに関するすべてのパラメータが含まれています。これらのパラメータは「データ書き出し」ダイアログボックス内で定義します。プロジェクトはディスクに保存され、読み込んで使用できるようになります。「データ書き出し」ダイアログボックスについての詳細は、この章の「データ読み込みとデータ書き出し」の節を参照してください。

有効なエクスポートプロジェクトを持つBLOBをプロジェクト引数に渡した場合、ユーザ操作なしに直接データ書き出しが実行されます。プロジェクトは「データ書き出し」ダイアログボックス内で前もって定義し、保存しておかなければいけません。これを行うには、2つの方法があります。

プロジェクトをディスクに保存後、**DOCUMENT TO BLOB** コマンドを使用してフィールドまたはBLOB変数にセットします。

空白のプロジェクト引数およびオプション引数*を指定した**EXPORT DATA**コマンドを実行し、プロジェクト引数のBLOBに保存します（下記参照）。この方法は、ディスク上からプロジェクトを読み込む必要はなく、データと共にプロジェクトを保存することができます。

オプションの引数*が指定されていれば、プロジェクト内に定義されたパラメータと共に「データ書き出し」ダイアログボックスを表示します。これは、パラメータの1つまたはそれ以上を変更できる可能性を持ちながら、前もって定義されたプロジェクトを使用できるようにするものです。さらに、「データ書き出し」ダイアログボックスを閉じた後に、プロジェクト引数は、「新しい」プロジェクトのパラメータを持つことができ、新しいプロジェクトをBLOBフィールドやディスク上などに保存することができます。

データ書き出しが正常に終了すると、システム変数OKは1になります。

参照

IMPORT DATA

命名セレクション

CREATE SELECTION FROM ARRAY

CREATE SELECTION FROM ARRAY(({テーブル;}レコード配列{;}セレクション名))

| 引数 | タイプ | 説明 |
|---------|------------------------|---|
| テーブル | テーブル | セレクションを作成するテーブル 省略した場合は現在のデフォルトテーブル |
| レコード配列 | 倍長整数配列 または ブール配列 | レコード番号の配列 または ブール配列 (True= レコードはセレクションに含める False= レコードはセレクションに含めない) |
| セレクション名 | 文字列 | 作成する命名セレクションの名前 省略した場合は、コマンドをカレントセレクションに適用する。 |

説明

このコマンドは、下記の方法で命名セレクションセレクション名を作成します：

絶対レコード番号の配列から、またはブールの配列から。

この場合、配列の値はテーブルの各レコードが作成する命名セレクションに含める (True)か含めない (False)かを示します。

セレクション名を省略した場合や空白を渡した場合は、コマンドはカレントセレクションに適用され、更新されます。

このコマンドで倍長整数配列を使用すると、配列の各要素は作成されるセレクション名内のレコードのレコード番号を表わします。レコード番号が不正確（作成されていないレコード）な場合、エラー-10503が発生します。

このコマンドでブール配列を使用すると、配列のN番目の要素がTrueかFalseにより、レコード番号Nが、作成するセレクション名に含まれるかどうかを表わしています。配列要素数は、テーブル内のレコード数と等しくなければなりません。配列数がレコード数よりも少ない場合は、配列によって定義されたレコードのみが判断対象となります。

注：ブール配列では、コマンドは配列要素0から配列要素N-1を使用します。

警告：命名セレクションが作成されてメモリ内にロードされます。したがって、このコマンドを実行する前に十分なメモリがあることを確認してください。

参照

CREATE SET FROM ARRAY

オブジェクトプロパティ

GET OBJECT RECT

GET OBJECT RECT({*;}オブジェクト;左;上;右;下)

| 引数 | タイプ | 説明 |
|--------|--------|--|
| * | | 指定された場合=オブジェクトはオブジェクトの名前（文字列）である。 省略された場合=オブジェクトは変数である。 |
| オブジェクト | オブジェクト | オブジェクト名（*が指定された場合）またはフィールドまたは変数（*が省略された場合） |
| 左 | 倍長整数 | オブジェクトの左の座標 |
| 上 | 倍長整数 | オブジェクトの上の座標 |
| 右 | 倍長整数 | オブジェクトの右の座標 |
| 下 | 倍長整数 | オブジェクトの下座標 |

説明

このコマンドは、引数*およびオブジェクトによって指定された、現在のフォームのオブジェクトの左、上、右、下の座標（ポイント）を返します。

オプション引数*を指定すると、オブジェクト引数がオブジェクト名（文字列）であることを示し、オプション引数*を省略すると、オブジェクト引数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または変数の参照（オブジェクトタイプがフィールドまたは変数のみ）を指定します。

オブジェクトにオブジェクト名としてワイルドカード(@)を使用し、複数のオブジェクトを指定すると、返される座標は関連する全オブジェクトで形成される長方形の座標になります。

注：バージョン6.5からは、文字列に含まれるワイルドカード文字(@)の取り扱い方を設定することができます。このオプションは、「オブジェクトプロパティ」コマンドに影響を与えます。詳細は、第2章「デザインモード」の「文字「@」の管理」を参照してください。

オブジェクトが存在しない場合やコマンドがフォーム内で呼び出されていない場合、座標(0;0;0;0)が返されます。

例題

"button"で始まるすべてのオブジェクトによって形成される長方形の座標を得たい場合を仮定します。

GET OBJECT RECT(*;"button@";left;top;right;bottom)

参照

MOVE OBJECT

MOVE OBJECT

MOVE OBJECT({*;}オブジェクト;水平移動;垂直移動;{*;}水平リサイズ{*;}垂直リサイズ){*;}{*;})

| 引数 | タイプ | 説明 |
|--------|--------|--|
| * | | 指定された場合=オブジェクトはオブジェクトの名前(文字列)である。省略された場合=オブジェクトは変数である。 |
| オブジェクト | オブジェクト | オブジェクト名(*が指定されている場合)または フィールドまたは変数(*が省略された場合) |
| 水平移動 | 倍長整数 | オブジェクトの水平移動距離(>0=右へ、<0=左へ) |
| 垂直移動 | 倍長整数 | オブジェクトの垂直移動距離(>0=下へ、<0=上へ) |
| 水平リサイズ | 倍長整数 | オブジェクトの水平方向のサイズ変更値 |
| 垂直リサイズ | 倍長整数 | オブジェクトの垂直方向のサイズ変更値 |
| * | | 指定されている場合=絶対座標 省略された場合=相対座標 |

説明

このコマンドは、*とオブジェクトの引数で定義された、現在のフォーム内のオブジェクトを、水平方向にピクセル、垂直方向にピクセル移動させます。また、(オプションで)オブジェクトを水平方向に水平リサイズピクセル、垂直方向にピクセル、サイズの変更をすることもできます。

移動とサイズ変更の方向は、水平移動および垂直移動引数に渡された値によります。

値が正であれば、オブジェクトは右および下へそれぞれ移動され、サイズ変更されません。

値が負であれば、オブジェクトは左および上へそれぞれ移動され、サイズ変更されません。

最初のオプション引数*を指定すると、オブジェクト引数がオブジェクト名(文字列)であることを示し、最初のオプション引数*を省略すると、オブジェクト引数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または変数の参照(オブジェクトタイプがフィールドまたは変数のみ)を指定します。

オブジェクトにオブジェクト名としてワイルドカード「@」を使用し、複数のオブジェクトを指定すると、関連する全オブジェクトが移動またはサイズが変更されます。

注：バージョン6.5からは、文字列に含まれるワイルドカード文字「@」の取り扱い方を設定することができます。このオプションは、「オブジェクトプロパティ」コマンドに影響を与えます。詳細は、第2章「デザインモード」の「文字「@」の管理」を参照してください。

デフォルトでは、水平移動、垂直移動、水平リサイズ、垂直リサイズの値は、オブジェクトの以前の位置からの相対的な値です。引数が絶対位置を表わすようにしたい場合は、最後のオプションの引数*を渡します。

例題

1. 下記のコードは、"button_1"を右に10ピクセル上に20ピクセル移動させ、幅を30ピクセル高さを40ピクセルにサイズ変更します。

MOVE OBJECT (*;"button_1";10;-20;30;40)

2. 下記のコード = は、"button_1"を次の座標に移動します：(10;20)(30;40)

MOVE OBJECT(*;"button_1";10;20;30;40;*)

参照

GET OBJECT RECT

ピクチャ

PICT TO GIF

PICT TO GIF(ピクチャ;GIF 用 BLOB)

| 引数 | タイプ | 説明 |
|------------|------|-----------------------|
| ピクチャ | ピクチャ | ピクチャタイプのフィールドまたは変数 |
| GIF 用 BLOB | BLOB | GIF タイプの画像をセットする BLOB |

説明

このコマンドは、変数またはフィールドに保存されている（PICTタイプの）ピクチャから、GIFフォーマットの画像を作成します。

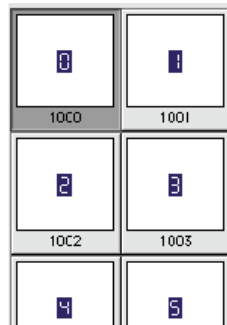
ピクチャタイプの変数またはフィールドをピクチャに、BLOBタイプの変数またはフィールドをblobGIFに渡します。コマンドの実行後、GIF用BLOBの内容はGIFフォーマットの画像になります。

注：GIFフォーマットは256色以上はサポートされていません。元のPICTピクチャがそれ以上の色数の場合は、失われる色があります。このコマンドで作られるGIF画像は色に応じて最適化され、これは87a(opaque)およびnormal(not interlaced)タイプのもので。

blobGIFの画像を**BLOB TO DOCUMENT**コマンドを使ってファイルに保存でき、またそれをWeb上に公開することもできます。

変換が成功したらシステム変数OKが1になります。そうでない場合は0になります。

接続カウンターを表示するGIF画像を作成する場合を想定してみます。データベースのピクチャライブラリー内に、すべての番号をピクチャとして登録しておきます。



例題

1. On Web Connection データベースメソッドで、次のコードを記述します。

```
If(Web Context)
:
Else
  C_BLOB ($blob)
  Case of
  :
  ¥($1="/4dcgi/counter")      `GIF counter の生成
    `スタティックホームページを送っている間に、4D がこの URL を検出すると
    `GIF カウンターを発生させる
  $blob:=gifcounter (<>nbHits) `GIF picture を得る
    `<>nbHits 変数は接続数
  SEND HTML BLOB ($blob;"image/gif")
    `ピクチャを挿入してブラウザへ送る
  :
End case
End if
```

2. gifcounter メソッドを示します。

```
C_LONGINT($1)
C_IMAGE($img)
C_BLOB($0)
If ($1=0)
  $ndigits:=1
Else
  $ndigits:=1+Length(String($1))
End if
If ($ndigits<5)
  $ndigits:=5
End if
$div:=10^($ndigits-1)
For ($i;1;$ndigits)
  $ref:=Int($1/$div)%10
  GET PICTURE FROM LIBRARY($ref+1000;picture)
  $img:=$img+picture
  $div:=$div/10
End for
PICT TO GIF($img;$0)
```

ページを Web ブラウザに送る時、4D は下記のタイプの GIF ピクチャを表示します。



プロセス（通信）

Test semaphore

Test semaphore(セマフォ) ブール

| 引数 | タイプ | 説明 |
|------|-----|------------------------------------|
| セマフォ | 文字列 | テストするセマフォの名前 |
| 戻り値 | ブール | True=セマフォが存在する False=セマフォが存在しない |

説明

この関数は、セマフォの存在をテストします。

Semaphore 関数と **Test semaphore** 関数の違いは、**Test semaphore** はセマフォが存在しない場合にはそのセマフォを作成しないということです。セマフォが存在している場合、関数はTrueを返します。そうでない場合はFalseを返します。

例題

下記の例は、セマフォを変更せずにプロセスの状態（この場合は、コードを変更している間）を知ることができるものです。

Open window (x1;x2;y1;y2;-Palette window)

Repeat

If (**Test semaphore**("Encrypting code"))

GOTO XY(\$x3;\$y3)

MESSAGE("暗号コードが変更されている")

Else

GOTO XY(\$x3;\$y3)

MESSAGE("暗号コードの変更が認証された")

End if

Until(StopInfo)

CLOSE WINDOW

参照

Semaphore

プロセス

REGISTER CLIENT

REGISTER CLIENT(クライアント名{;ピリオド}{;{*})

| 引数 | タイプ | 説明 |
|---------|------|---------------------------|
| クライアント名 | 文字列 | 4D クライアントセッションの名前 |
| ピリオド | 倍長整数 | 4D Server への問い合わせ間隔 (秒単位) |
| * | * | ローカルプロセス |

説明

このコマンドは、4D クライアントステーションを、クライアント名内で指定した名前で 4D Server に「登録」し、他のクライアントもしくは 4D Server (保存されたメソッドを使って) が、**EXECUTE ON CLIENT** コマンドを使ってメソッドを実行できるようにします。一旦登録されると、4D Server は他のクライアント用に 1 つまたはそれ以上のメソッドを実行することができます。

注: 「データベースプロパティ」ダイアログの「開始時にクライアント登録」オプションを使って、4D Server に接続するクライアントステーションを自動的に登録することができます (詳細は、第 2 章「デザインモード」の「スタートアップ時にクライアントを登録する」(4D Server のみ) を参照してください)。

このコマンドが実行されると、クライアントステーション上にクライアント名という名のプロセスが作成されます。このプロセスは、**UNREGISTER CLIENT** コマンドによってのみアポートが可能です。

オプションの * 引数を渡すと、作成されるプロセスはローカルプロセスになり、4D は自動的にプロセス名の始めにドルマーク(\$)を付け加えます。そうでない場合は、グローバルプロセスです。

コマンドの実行後、クライアントステーションは、他の 4D Client または 4D Server 自身が呼び出しをしていないかどうかを見るために、定期的に 4D Server に問い合わせます。デフォルトではこの問い合わせが 2 秒ごとに行われ、この間隔はピリオドを変更することによって変えることができます。最小値は 1 秒です。

注: 4th Dimension のシングルユーザバージョンで、このコマンドは何も行ないません。

一旦コマンドが実行されると、実行中の 4D Client の名前または 4D Server へ問い合わせ間隔を変更することはできません。これを行うには、**UNREGISTER CLIENT** コマンドを呼び出し、再度 **REGISTER CLIENT** コマンドを呼び出します。

注: 複数の 4D クライアントが同じ登録名を持つことができます。

4D Clientが正しく登録されると、システム変数OKは1になります。4D Clientが既に登録されていると、コマンドは何も行わずOKは0になります。

例題

下記の例では、クライアントワークステーション同士の間で通信を行なうことができる、小さなメッセージシステムを作成してみます。

1. このメソッドRegistrationは4D Clientを登録して、他の4D Clientからのメッセージを受け取ることができるようにします。

　`他の名前で登録する前に、登録の解除が必要です

UNREGISTER CLIENT

Repeat

　vPseudoName:=**Request**("登録名 : ","User","OK","Cancel")

Until ((OK=0) | (vPseudoName # ""))

If (OK=0)

　... `登録しない場合の処理

Else

REGISTER CLIENT(vPseudoName)

End if

2. 次の指示は、登録された4D Clientのリストを得ることができるようにするものです。これは、On Startup データベースメソッド内に置くことができます。

　PrClientList:=**New process**("4D Client List";32000;"List of registered clients")

3. 4D Client List メソッドは、登録している全4D Clientの登録名リストを入手し、表示リストを更新します。

If (Application type=4D Client)

　`下記のコードはClient & Server モードでのみ有効です

　\$Ref:=**Open window**(100;100;300;400;-(Palette window+Has window title);
　"List of registered clients")

Repeat

GET REGISTERED CLIENTS(\$ClientList;\$ListeCharge)

　　`\$ClientList 内の登録されたクライアントで更新する

ERASE WINDOW(\$Ref)

GOTO XY(0;0)

For (\$p;1;Size of array(\$ClientList))

MESSAGE(\$ClientList{\$p}+Char(Carriage return))

End for

　　`1 秒延滞

DELAY PROCES(Current process;60)

Until (False)　　`無限ループ

End if

4. 下記のメソッドは、登録している他の4D Clientにメッセージを送ります。これは、送られた4D ClientでDisplay_Messageメソッドを呼び出します（下記参照）。

```
$Addressee:=Request("メッセージ送信先：","")
`On startup データベースメソッドで作成された表示リストに
`表示されている名前を入力します
If (OK # 0)
    $Message:=Request("送信メッセージ：") `メッセージ入力
    If (OK # 0)
        EXECUTE ON CLIENT($Addressee;"Display_Message";
                           $Message) `メッセージを送る
    End if
End if
```

5. 最後に、クライアントステーションが他の4D Clientから見えず、メッセージも受け取れなくなるようにします。

UNREGISTER CLIENT

参照

EXECUTE ON CLIENT、UNREGISTER CLIENT、GET REGISTERED CLIENT

UNREGISTER CLIENT

UNREGISTER CLIENT

| 引数 | タイプ | 説明 |
|----|-----|----|
|----|-----|----|

このコマンドには、引数はありません。

説明

このコマンドは、クライアントステーションの登録を解除します。クライアントは **REGISTER CLIENT** コマンドによって既に登録されているものでなければなりません。

注：4D Client は、ユーザがアプリケーションを終了すると、自動的に登録を解除します。

クライアントステーションが前もって登録されていなかったり、コマンドをシングルユーザ 4th Dimension で実行しても、コマンドは何も行ないません。

クライアントの登録が正しく解除されると、システム変数 OK は 1 になり、クライアントが登録されていないと、OK は 0 になります。

例題

REGISTER CLIENT コマンドの例を参照してください。

参照

REGISTER CLIENT、GET REGISTERED CLIENTS

GET REGISTERED CLIENTS

GET REGISTERED CLIENTS(クライアントリスト;メソッド)

| 引数 | タイプ | 説明 |
|-----------|--------|---------------------|
| クライアントリスト | テキスト配列 | 保存された 4D Client の配列 |
| メソッド | 倍長整数配列 | 実行待ちメソッド数の配列 |

説明

このコマンドは2つの配列を作成します。

クライアントリストは、**REGISTER CLIENT** コマンドを使って「登録された」クライアントの配列となります。

メソッドは、各クライアントの「ワークロード」の配列となります。ワークロードは、**EXECUTE ON CLIENT** コマンドを呼び出すことによって発生した、4D Client がこれから実行しなければならないメソッドの数です（詳細は、「**EXECUTE ON CLIENT**」コマンドの説明を参照してください）。

正常に処理された場合、システム変数 OK は 1 になります。

例題

1. すべての登録されたクライアントの配列と、まだ実行されずに残っているメソッド数の配列の作成を考えてみます。

```
ARRAY TEXT($clients;0)
ARRAY LONGINT($methods;0)
GET REGISTERED CLIENTS($clients;$methods)
```

2. **REGISTER CLIENT** コマンドの例を参照してください。

参照

REGISTER CLIENT、EXECUTE ON CLIENT

EXECUTE ON CLIENT

EXECUTE ON CLIENT(クライアント名;メソッド名{;引数1;...;引数N})

| 引数 | タイプ | 説明 |
|---------|-----|---------------|
| クライアント名 | 文字列 | 4D Clientの登録名 |
| メソッド名 | 文字列 | 実行するメソッドの名前 |
| 引数1...N | | メソッドの引数 |

説明

このコマンドは、名前がクライアント名である登録されている4D Clientで、引数1...引数Nを引数としたメソッド名メソッドを実行します。4D Clientの登録名は**REGISTER CLIENT**コマンドで定義されます。

このコマンドは、4D Clientまたは4D Serverのストアードプロシージャによって呼び出すことができます。

メソッドが1つまたはそれ以上の引数が必要な場合、メソッドの名前の後に渡してください。

指定した4D Client上でのメソッドの実行は、指定した4D Client上で自動的に作成されたグローバルプロセス内で行われます。そのプロセス名は4D Clientの登録名にもなっています。

このコマンドが、同じ4D Clientに対し複数の4D Clientから連続的に呼び出されると、実行の順序がスタックされます。したがって、メソッドは非同期モードで、次から次へと処理されます。スタックされたメソッドが増えると、4D Clientのワークロードも大きくなってしまいます。**GET REGISTERED CLIENTS**コマンドを使用して、各クライアントのワークロードの状態を知ることができます。

注：実行順序のスタックは、**UNREGISTER CLIENT**コマンドを使って4D Clientが登録解除されない限りは、変更することも止めることもできません。

登録された複数の4D Clientに対し、同じメソッドを同時に実行させることができます。これを行うには、クライアント名引数にワイルドカード(@)を使用します。

システム変数OKは、4D Serverがメソッドの実行要求を正しく受け取った場合、1になりますが、これはメソッドが4D Clientによって正しく実行されたということを保証するものではありません。

例題

1. 「GenerateNums」メソッドを「Client1」クライアントステーション上で実行させたい場合には、

```
EXECUTE ON CLIENT("Client1";"GenerateNums";12;$a;"Text")
```

2. すべてのクライアントに"EmptyTemp"メソッドを実行させたい場合には、

```
EXECUTE ON CLIENT("@";"EmptyTemp")
```

3. **REGISTER CLIENT** コマンドの例を参照してください。

参照

REGISTER CLIENT、UNREGISTER CLIENT、GET REGISTERED CLIENTS

Process aborted

Process aborted ブール

引数 タイプ 説明
このコマンドには、引数はありません。

戻り値 ブール True= プロセスがアボートされようとしている
False= プロセスはアボートされようとしていない

説明

この関数は、この関数を呼び出したプロセスが不意に中断されようとしている場合、Trueを返します。これは、関数の実行が正常に完了しなかったことを意味します。例えば、これはQUIT 4Dを呼び出した後に発生します。

このコマンドは、コンパイルモードでのみ、Webサーバ上のプログラミングの特別なケースとして使用できます。While...End while（例を参照）のようなループを使ってWebページを送るメソッドを使用する場合、Webサーバの構造として、Webブラウザのタイムアウト（指定された時間の終了）が発生しても、ループを停止できません。Webプロセスが閉じられていないので、Webライセンスは使用されたままとなります。

ループの終了判断として使用している**Process aborted**コマンドは、タイムアウトが発生するとTrueを返すため、ループは中断されてWebライセンスが開放されます。

例題

1. HTML ページを送るために使用できるメソッドを紹介します。コンパイルドモードでは、このループはタイムアウトが発生しても中断することができません。

```
While (True)
  SEND HTML FILE (HTMLFile)
End while
```

2. **Process aborted**関数を使用すると、同じ動作を可能にすると共に、タイムアウトが発生するとループを抜け、Webライセンスを開放することができます。

```
While (Not (Process aborted))
  SEND HTML FILE (HTMLFile)
End while
```

クエリ

QUERY WITH ARRAY

QUERY WITH ARRAY(インデックスフィールド;配列)

| 引数 | タイプ | 説明 |
|-------------|-------|-----------------------|
| インデックスフィールド | フィールド | 比較の為に使用されるインデックスフィールド |
| 配列 | 配列 | 検索された値の配列 |

説明

QUERY WITH ARRAY コマンドは、インデックスフィールドの値が配列要素の内容と等しい、すべてのレコードを検索し、新しいカレントセクションを作成します。

このコマンドは、速やかに、かつ簡単に複数の値に対する検索を可能にします。

注：

- ・ このコマンドは、インデックスフィールドにのみ有効です。タイプがテキスト、ピクチャ、サブフィールドまたはBLOBのフィールドには使用できません。
- ・ 倍長整数タイプの配列は、時間タイプのフィールドと互換性のあることを覚えておいてください。

例題

下記の例は、フランスとアメリカの顧客を検索するものです。

ARRAY STRING (2;SearchArray;30)

SearchArray{1}:="FR"

SearchArray{2}:="US"

QUERY WITH ARRAY ([Clients]Country;SearchArray)

Find index key

Find index key (インデックスフィールド;値) 倍長整数

| 引数 | タイプ | 説明 |
|-------------|-------|-----------------------------|
| インデックスフィールド | フィールド | 検索を実行するインデックスフィールド |
| 値 | | 検索する値 見つかった値 |
| 戻り値 | 倍長整数 | レコード番号 レコードが何も見つからなければ-1 |

説明

この関数は、そのインデックスフィールドが値と等しい最初のレコード番号を返します。

何もレコードが見つからなければ、**Find index key**は-1を返します。

この関数を呼び出した後、値には見つかった値が挿入されます。これは、文字列フィールド上でワイルドカード「@」を使って検索し、見つかった値を知ることができるようにします。

この関数は、カレントセレクションまたはカレントレコードを変更しません。

インデックスのみを使用しているので速く、特にレコード入力中に重複データの入力を防ぐのに役立ちます。

例題

オーディオCD用のデータベースで、レコード入力中に、既に登録されている歌手かどうかを確認したいと仮定します。同姓同名も存在するので[Singer]Nameフィールドを重複不可にせず、入力フォームで[Singer]Nameフィールドのオブジェクトメソッドに下記のコードを書くことにします。

```
If (Form event=On Data Change)
  $RecNum:=Find index key([Singer]Name;[Singer]Name)
  If ($RecNum # -1)      `この名前が既に登録されている場合
    CONFIRM("同名のシンガーが既に存在します。レコードを見ますか?";
    "はい";"いいえ")
    If (OK=1)
      GOTO RECORD([Singer];$RecNum)
    End if
  End if
End if
End if
```

レコード

Is new record

Is new record({テーブル}) ブール

| 引数 | タイプ | 説明 |
|------|------|---|
| テーブル | テーブル | 検査するレコードの属するテーブル。この引数が省略されている場合はデフォルトテーブル |
| 戻り値 | ブール | 未登録の新規レコードの場合 True、そうでなければ False |

説明

この関数は、指定されたテーブルのカレントレコードが未登録の新規レコードの場合に True を返します。

互換性について：

Record number関数が-3を返すかどうかで同じ情報を得ることができますが、**Record number**の代りに**Is new record**を使用することを強くお勧めします。実際、**Is new record**関数は4th Dimensionの将来のバージョンとの、より優れた互換性を保証します。

例題

下記の2つの方法は同一のものです。コードが4Dの将来のバージョンとの互換性を保つため、2番目の方法を強く推奨します。

```
If (Record number([Table])=-3)      `この方法は推奨しません
    `...
End if
If (Is new record([Table]))          `この方法を使用してください
    `...
End if
```

参照

Record number

Is record loaded

Is record loaded(*{table}*) ブール

| 引数 | タイプ | 説明 |
|------|------|---|
| テーブル | テーブル | 検査するレコードの属するテーブル。この引数が省略されている場合はデフォルトテーブル |
| 戻り値 | ブール | レコードがロードされていればTrue、それ以外はFalse |

説明

この関数は、指定テーブルのカレントレコードがカレントプロセス内にロードされていればTrueを返します。

例題

「次レコード」または「前レコード」の自動動作を使用する代わりに、これら2つのボタン用にオブジェクトメソッドを書き、これらの作業を改善させることができます。「Next」ボタンは、セレクションの最後のレコードであれば、セレクションの最初のレコードを表示します。「Previous」ボタンは、セレクションの最初のレコードであれば、セレクションの最後のレコードを表示します。

　　`"Previous"ボタンのオブジェクトメソッド(自動動作無し)

```
If (Form event=On Clicked)
    PREVIOUS RECORD([Group])
    If (Not(Is record loaded([Group])))
        GOTO SELECTED RECORD([Group];Records in selection([Group]))
        `セレクションの最後のレコードに移行
    End if
End if
```

　　`"Next"ボタンのオブジェクトメソッド(自動動作無し)

```
If (Form event=On Clicked)
    NEXT RECORD([Group])
    If (Not(Is record loaded([Group])))
        GOTO SELECTED RECORD([Groups];1)
        `セレクションの最初のレコードに移動
    End if
End if
```

セレクション

HIGHLIGHT RECORDS

HIGHLIGHT RECORDS ({セット名})

| 引数 | タイプ | 説明 |
|------|-----|-------------------------------------|
| セット名 | 文字列 | 反転表示させるレコードのセット引数が省略されている場合はUserSet |

説明

このコマンドは、出力フォーム内で指定された複数のレコードを反転表示させることができます。この操作は、マウスまたはshiftキーを押しながらクリック、あるいはCtrlキーを押しながらクリック（Macintoshの場合はcommandキーを押しながらクリック）によって、リスト上のレコードを選択するのと同じものです。

「選択された」レコードは反転表示されますが、カレントセレクションは変更されません。

注：UserSetセットはレコードリストの再表示で更新されます。すなわち、呼び出したメソッドの全体を実行した後であり、HIGHLIGHT RECORDSを実行した直後ではありません。

引数<セット名>に有効な名前を渡すと、そのセットのレコードにコマンドが適用されます。

引数<セット名>を省略すると、コマンドは現在のUserSetに属するレコードだけに適用されます。

MODIFY SELECTION コマンドによって表示される出力フォーム内で、カレントセレクションを変更することなく、ユーザが検索を実行できるようにしたい場合は、フォーム内にSearchボタンを置き、押された時に下記のメソッドを実行します。

SET QUERY DESTINATION(Into Set;"UserSet")

QUERY

SET QUERY DESTINATION(Into Current Selection)

HIGHLIGHT RECORDS

ユーザがボタンをクリックすると、標準の「検索」ダイアログボックスが現れ、検索が実行された後、カレントセレクションを変更することなく、見つかったレコードを反転表示することができます。

セット

CREATE SET FROM ARRAY

CREATE SET FROM ARRAY(({テーブル;}レコード配列;{セット名})

| 引数 | タイプ | 説明 |
|--------|------------------------|---|
| テーブル | テーブル | セットを作成するテーブル 引数が省略されている場合はデフォルトテーブル |
| レコード配列 | 倍長整数配列 または ブール配列 | レコード番号の配列、またはブール配列 (True=レコードをセットに含める、 False=レコードをセットに含めない) |
| セット名 | 文字列 | 作成するセット名 省略された場合は UserSet に適用される |

説明

このコマンドは、下記の方法でセット名を作成します。

テーブルの絶対レコード番号の配列から、またはブール配列から：

この場合、配列の値は、テーブル内の各レコードをセット名に含める(True)か含めない(False)かを表わします。

このコマンドを使って倍長整数配列をレコード配列へ渡す時、配列要素の内容はセット名に含めるレコードのレコード番号を表わしています。無効なレコード番号を含んでいる場合（例えば、まだレコードが作成されていない場合）、エラー-10503が発生します。

このコマンドを使ってブール配列をレコード配列へ渡す時、配列のN番目の要素は、レコード番号Nのレコードをセット名に含める(True)か含めない(False)かを表わします。通常、配列のサイズはテーブル内のレコードの数と等しくなければなりません。レコードの数よりも配列が少ない場合は、配列に定義されたレコードだけを対象に判断します。

注：ブール配列の場合、このコマンドは配列要素0から配列要素N-1までを使用します。

セット名引数を省略したり空白を渡すと、コマンドは UserSet に適用されます。

参照

CREATE SELECTION FROM ARRAY

ストラクチャアクセス

SET DATABASE PARAMETER

SET DATABASE PARAMETER({テーブル;}セレクトタ;値)

| | | |
|-------|------|---------------------------------------|
| 引数 | タイプ | 説明 |
| テーブル | テーブル | 属性を設定するテーブル 引数が省略されている場合はデフォルトテーブル |
| セレクトタ | 倍長整数 | 変更するデータベース属性コード |
| 値 | 倍長整数 | 属性の値 |

説明

このコマンドは、カレントプロセス用に4Dデータベース内部の様々な属性を変更することができます。

セレクトタは、変更するデータベースの属性コードを指定します。4th Dimensionは「データベース属性」のカテゴリー内に、前もって定義されている下記のような定数があります。

| | | |
|---------------------------|------|---|
| 定数 | タイプ | 値 |
| Seq Order Ratio | 倍長整数 | 1 |
| Seq Access Optimization | 倍長整数 | 2 |
| Seq Distinct Values Ratio | 倍長整数 | 3 |
| Index Compacting | 倍長整数 | 4 |
| Seq Query Select Ratio | 倍長整数 | 5 |
| Minimum Web Process | 倍長整数 | 6 |
| Maximum Web Process | 倍長整数 | 7 |
| Web Conversion | 倍長整数 | 8 |

値は、属性の値を指定します。値の内容は変更しようとする属性によって違います。

セレクトで指定する可能性のある値を示します。

| セレクト | 値 | 内容 |
|------|-----------------------------------|--|
| 1 | 0 100,000 | レコード（リレートされるレコードも含んだ合計数）の選択率。その率以下ではソートがシーケンシャルモードで実行されます。この率は、100,000分の1単位で表わされます。デフォルト値は9,000(=9%)です。 |
| 2 | 0 または1 0：最適化されず 1：最適化する | シーケンシャルアクセス（配列のソート、検索、選択）用の最適化モード。最適化モードでは、4Dはディスクからの多くのレコードを一度に読もうとしますが、これらをキャッシュ内には置きません。このモードは、キャッシュのサイズが低いからです。デフォルトでは、値は1になります（最適化モード）。詳細は、第6章「最適化」を参照してください。 |
| 3 | 0 100,000 | レコード（リレートされるレコードも含んだ合計数）の選択率。その率以下では DISTINCT VALUES コマンドがシーケンシャルモードで実行されます。この率は、100,000分の1単位で表わされます。デフォルト値は0です。 |
| 4 | 0 または1 0：no、1：yes | インデックスページ圧縮の起動または非起動。デフォルトでは値は1(インデックスは必要であればコンパクト化される)です。詳細は、第6章「最適化」を参照してください。 |
| 5 | 0 100,000 | レコード（リレートされるレコードも含んだ合計数）の選択率。その率以下では QUERY SELECTION コマンドがシーケンシャルモードで実行されます。この率は、100,000分の1単位で表わされます。デフォルト値は0です。 |
| 6 | 0 32,767 | 非コンテキストモード内に保持するWebプロセスの最少数 ⁽¹⁾ 。デフォルト値は0です。 |
| 7 | 0 32,767 | 非コンテキストモード内に保持するWebプロセスの最大数 ⁽¹⁾ 。デフォルトは0です。 |
| 8 | 1または2 1:6.0.x モード 2:6.5 モード | 4D フォームのWeb変換モード。デフォルトで値は2(6.5モード)です。 |

1. Webサーバが、非コンテキストモードでプロセスの再利用をするために、4DはWebプロセスを5秒間延滞し、次に起こりうるHTTPリクエストの実行のために待機させます。能力の面で言えば、各問い合わせに新しいプロセスを作成するよりも、この原理はずっと利点の多いものです。Webプロセスが再利用されると、もう一度5秒間延滞させられます。5秒以内に何のリクエストも発生しない場合、Webプロセス数が指定した最少数でなければプロセスはアボートされ、最少数に達した場合は再度延滞させられます。

これらの引数は、リクエストの数やメモリなどに応じて、Webサーバの機能を調整できるようにするものです。

参照

Get database parameter

Get database parameter

Get database parameter({テーブル;}セクタ) 倍長整数

| 引数 | タイプ | 説明 |
|------|------|--|
| テーブル | テーブル | 属性の値を得るテーブル。 この引数が省略されている場合はデフォルトテーブル |
| セクタ | 倍長整数 | データベース属性コード |
| 戻り値 | 倍長整数 | 属性の値 |

説明

この関数は、カレントプロセス用の4Dデータベース属性の値を読み込むことができます。
セクタ引数は、読み込む属性を指定します。4th Dimensionは「データベース属性」の
カテゴリー内に、前もって定義されている下記のような定数があります。

| 定数 | タイプ | 値 |
|---------------------------|------|---|
| Seq Order Ratio | 倍長整数 | 1 |
| Seq Access Optimization | 倍長整数 | 2 |
| Seq Distinct Values Ratio | 倍長整数 | 3 |
| Index Compacting | 倍長整数 | 4 |
| Seq Query Select Ratio | 倍長整数 | 5 |
| Minimum Web Process | 倍長整数 | 6 |
| Maximum Web Process | 倍長整数 | 7 |
| Web Conversion Mode | 倍長整数 | 8 |
| Database Cache Size | 倍長整数 | 9 |

この関数によって返される値については、**SET DATABASE PARAMETER** コマンドの説明を参照してください。

参照

SET DATABASE PARAMETER

システムドキュメント

Select folder

Select folder({メッセージ}) 文字列

| 引数 | タイプ | 説明 |
|-------|-----|--------------------------|
| メッセージ | 文字列 | 表示される「フォルダ選択」ダイアログ内のタイトル |
| 戻り値 | 文字列 | 選択されたフォルダへのアクセスパス |

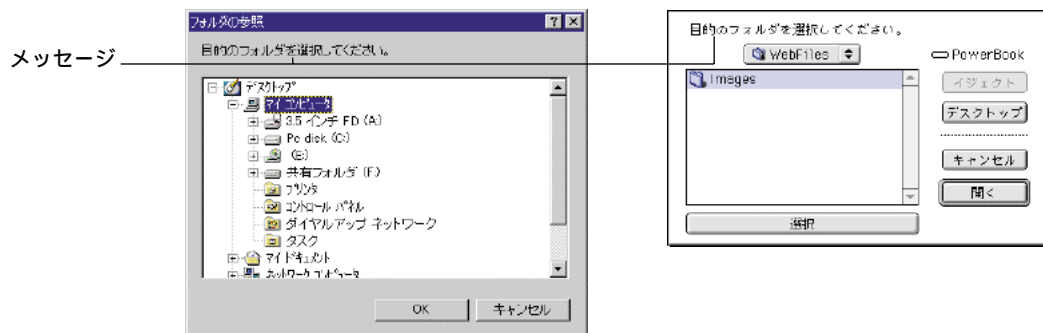
説明

この関数は、「フォルダ選択」ダイアログボックスを表示し、選択したフォルダのフルパスを返します。

注：この関数は、4Dのカレントフォルダを変更しません。

Select folder コマンドはワークステーションのボリュームおよびフォルダ内をナビゲートするための標準のダイアログボックスを表示します。

オプションの引数メッセージを指定すると、ダイアログボックス内に表示されます。



ユーザはフォルダを選択して、「OK」ボタンをクリック（Windowsの場合）するか、または選択ボタンをクリック（Macintoshの場合）すると、フォルダへのフルパスが関数値として返されます。

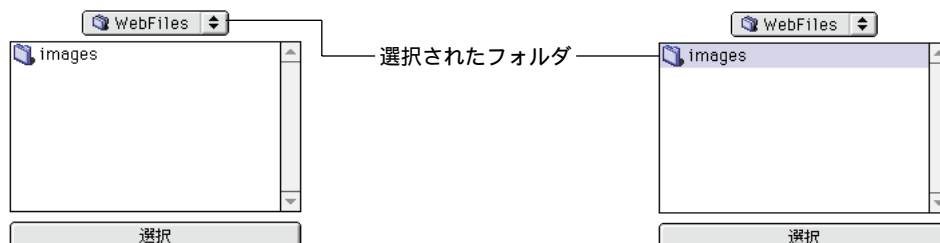
Windowsでのパスは、下記のフォーマットで返されます。

"C:¥Folder1¥Folder2¥SelectedFolder¥"

Macintoshでのパスは、下記のフォーマットで返されます。

"Hard Disk:Folder1:Folder2:SelectedFolder:"

注：Macintoshでは、ダイアログボックス内でフォルダの名前を選択しているかどうかで、帰ってくるパスは違うものになる場合があります。



4D Server：この関数は、クライアントワークステーションに接続されているボリュームを見ることができるようにするものです。ストアードプロシージャからこの関数を呼び出すことはできません。

ユーザがダイアログボックスを受け入れるとシステム変数OKは1になります。ユーザがキャンセルボタンをクリックするとシステム変数OKは0になり、関数は空白を返します。

注：Windowsでは、ユーザが何らかの誤ったエレメント（「ワークステーション」や「ごみ箱」など）を選択すると、ユーザがダイアログボックスを受け入れても、システム変数OKは0になります。

例題

下記の例では、フォルダを選択し、ピクチャライブラリ内のピクチャを保存します。

\$PictFolder:=Select folder("ピクチャを保存するフォルダを選択してください")

PICTURE LIBRARY LIST (pictRefs;pictNames)

For (\$n;1;Size of array(pictNames))

\$vRef:=Create document(\$PictFolder+pictNames{\$n};"PICT")

If (OK=1)

GET PICTURE FROM LIBRARY(pictRefs{\$n};\$vStoredPict)

SAVE PICTURE TO FILE(\$vRef;\$vStoredPict)

CLOSE DOCUMENT(\$vRef)

End if

End for

システム環境

LOG EVENT

LOG EVENT(メッセージ{重要度})

| 引数 | タイプ | 説明 |
|-------|-----|--------------|
| メッセージ | 文字列 | メッセージの内容 |
| 重要度 | 整数 | メッセージの重要度レベル |

注：この機能はWindowsNTでのみ可能です。

説明

バージョン6.5は、WindowsNTの"Log events"を利用できるようになりました。このログファイルは、実行されているアプリケーションから来るメッセージを受け取り、保存します。したがって、ワークセッションの動向を管理できるようになります。詳細は、WindowsNTのマニュアルを参照してください。

注：この機能を使用するには、WindowsNTの"Log events"サービスを開始する必要があります。

LOG EVENT コマンドは、WindowsNTの"Log events"に現れるカスタムメッセージを追加することができますようにします。

メッセージに、ログイベント内に書くメッセージを渡します。

メッセージに重要度を付けることができ、ログイベントを読む、または解析する助けになります。重要度には、「情報」、「警告」および「エラー」の3つのレベルがあります。重要度引数には、メッセージの重要度レベルを設定します。4th Dimensionには、前もって定義された次のような定数があります。これらは"WindowsNT Log Events"カテゴリーに置かれています。

| 定数 | タイプ | 値 |
|-----------------------------|-----|---|
| Information Message(デフォルト値) | 整数 | 0 |
| Warning Message | 整数 | 1 |
| Error Message | 整数 | 2 |

重要度が省略されたり正しくない値を指定すると、デフォルト値(0)が使用されます。

例題

データベースが開かれた時の情報をログを取りたい場合は、下記のコードをOn Startupデータベースメソッド内に書きます。

LOG EVENT("請求書DBオープン")

データベースが開かれるたびに、この情報はWindowsNTのログイベントに重要度レベルが0として書き込まれます。

SEND HTML BLOB

SEND HTML BLOB(blob;タイプ{;非コンテキスト})

| 引数 | タイプ | 説明 |
|---------|------|--|
| blob | BLOB | ブラウザへ送る内容を持つBLOB |
| タイプ | 文字列 | BLOB内のデータのデータタイプ |
| 非コンテキスト | ブール | True=非コンテキストモードへの切り替え False=コンテキストモードへの切り替え |

説明

このコマンドは、blobの内容をブラウザへ送ります。

blobに含まれるデータのタイプはタイプによって示されます。この引数には下記のタイプから、1つを指定します。

type=空白：この場合、それ以上の情報は何も供給する必要はありません。ブラウザはBLOB内の内容を解釈しようとします。

type=ファイル拡張子(例：".HTM"、".GIF"、".JPEG"など)：この場合、BLOB内に含まれるデータのMIMEタイプを拡張子を使って指定します。BLOBの内容は拡張子に応じて解釈されますが、ブラウザが正確に解釈できるように拡張子は標準のものでなければなりません。

type=Mime/Type(例："text/html"、"image/tiff"など)：この場合、BLOB内に含まれるデータのMIMEタイプを直接指定します。標準タイプの他に、固有のドキュメントをイントラネット経由で送るためにカスタムMIMEタイプを渡すこともできます。これを行うには、送られるタイプを認識し、適切なアプリケーションが開けるようにブラウザを設定するだけです。この場合、typeに渡す形式は"application/x-[TypeName]"です。クライアントワークステーションのブラウザでは、このタイプを"アプリケーション起動"動作に関連付けます。この方法で、**SEND HTML BLOB** コマンドはすべてのタイプのドキュメントを送ることが可能となり、イントラネットクライアントは関連するアプリケーションを自動的に開くことができるようになります。

注：BLOBがタイプ"text/html"(.htm、.html、.shtm、.shtml)である場合、HTMLファイルとして4Dによって内容を解析され必要な処置が施されます。

最も一般的な MIME タイプのリストです。

| 拡張子 | Mime/Type |
|--------|--------------------------|
| .htm | text/html |
| .html | text/html |
| .shtml | text/html |
| .shtm | text/html |
| .css | text/css |
| .pdf | application/pdf |
| .rtf | application/rtf |
| .ps | application/postscript |
| .eps | application/postscript |
| .hqx | application/mac-binhex40 |
| .js | application/javascript |
| .txt | text/plain |
| .text | text/plain |
| .gif | image/gif |
| .jpg | image/jpeg |
| .jpeg | image/jpeg |
| .jpe | image/jpeg |
| .jfif | image/jpeg |
| .pic | image/pict |
| .pict | image/pict |
| .tif | image/tiff |
| .tiff | image/tiff |
| .mpeg | video/mpeg |
| .mpg | video/mpeg |
| .mov | video/quicktime |
| .moov | video/quicktime |
| .aif | audio/aiff |
| .aiff | audio/aiff |
| .wav | audio/wav |
| .ram | audio/x-pn-realaudio |
| .sit | application/x-stuffit |
| .bin | application/x-stuffit |
| .z | application/x-zip |
| .zip | application/x-zip |
| .gz | application/x-gzip |
| .tar | application/x-tar |

非コンテキスト引数は、4D Webサーバに「コンテキストモード」から「非コンテキストモード」への切り替え、またはその逆の指示ができるようにするものです。

非コンテキストモードを使用するには非コンテキストへTrueを渡します。コンテキストモードを使用するにはFalseを渡します。引数が省略されると、デフォルトでコンテキストモードが使用されます。4D変数およびページ内の4DACTIONタグへの参照は、モードがなんであれ、常に解釈されます。

注：「非コンテキスト」モードのサポートはバージョン6.5で新しく追加された機能です。詳細は、第4章「Webサーバ」の「非コンテキストモード」参照してください。

例題

PICT TO GIFの例を参照してください。

Web Context

Web Context ブール

| 引数 | タイプ | 説明 |
|--------------------|-----|--------------------------------------|
| このコマンドには、引数はありません。 | | |
| 戻り値 | ブール | True= コンテキストモード False= 非コンテキストモード |

説明

この関数は、Web プロセスから呼び出されなければなりません。これは、Web の接続がコンテキストモード(True)で実行されているのか、それとも非コンテキストモード(False)で実行されているのかを返します。

注：Web プロセス以外のプロセスから呼び出されると、この関数は常に False を返します。

この関数の使用は、On Web Connection データベースメソッド内で行なうことをお勧めします（詳細は、第4章「Web サーバ」の「On Web Authentication データベースメソッド」を参照してください）

注：「非コンテキスト」モードのサポートはバージョン 6.5 で新しく追加された機能です。詳細は、第4章「Web サーバ」の「On Web Authentication データベースメソッド」を参照してください。

例題

On Web Connection データベースメソッドの例を示します。

If (Web Context)

WithContext (\$1;\$2;\$3;\$4;\$5;\$6)

Else

WithoutContext (\$1;\$2;\$3;\$4;\$5;\$6)

End if

参照

PROCESS PROPERTIES

WEB CACHE STATISTICS(ページ;ヒット数;使用率)

| 引数 | タイプ | 説明 |
|------|--------|-----------------------|
| ページ | テキスト配列 | キャッシュされている参照されたページの名前 |
| ヒット数 | 倍長整数配列 | 各ページのヒット数 |
| 使用率 | 数値 | キャッシュの使用率 |

このコマンドは、Webサーバのキャッシュ内にロードされている、ページについての情報を得ることができるようにするものです。そのため、これらの統計はスタティックホームページ、GIFピクチャ、JPEGピクチャ(100KB未満の)およびスタイルシート(.css)のみに適応されます。

注：4D Web サーバのキャッシュの設定に関する詳細は、第4章「Web サーバ」の「スタティックホームページ」を参照してください。

コマンドは、キャッシュされている参照されたページの名前でテキスト配列を作成します。倍長整数配列には各ページのヒット数がセットされ、使用率引数にはWebキャッシュの使用率がセットされます。

Web キャッシュの統計を表示するセミダイナミックページを作成したいと仮定します。このためには、"stats.shtm"と名付けられたスタティックHTML ページ内に、タグ<!--4D ACTION/STATS-->を置き（「4D ACTION タグ」を参照）、2つの4D 変数 vPage および vUsage を挿入します。

プロジェクトメソッドSTATSの内容です。

C_TEXT (\$1)

ARRAY TEXT (pages;0)

ARRAY LONGINT (hits;0)

C_LONGINT (vUsage)

WEB CACHE STATISTICS(pages;hits;vUsage)

vPages:=Char(1)

```
For ($i;1;Size of array(pages))
```

、キャッシュ内に存在する各ページ

```
vPages:=vPages+pages{$i}+"    " +String(hits{$i})+"<br>"
```

、ページの名前とHTMLコードを挿入

End for

```
SEND HTML FILE("stats.shtm")
```

1. 拡張子「.shtm」の付くページの内容は、常に4Dによって内容を解析され、必要な処理が施されます。詳細は、第4章「Webサーバ」の「セミダイナミックページ」を参照してください。

SEND HTTP REDIRECT

SEND HTTP REDIRECT(url{;*})

| 引数 | タイプ | 説明 |
|-----|-----|---|
| url | 文字列 | 新しいURL |
| * | | 指定されている場合=4DはURLをエンコードしない 省略されている場合=4DはURLをエンコードする |

説明

このコマンドは、URLのリダイレクトを可能にします。

url引数には、リダイレクトした新しいURLが含まれています。この引数がファイルへのurlである場合、非コンテキストモードと同じファイル参照方法でなければなりません。例えば、

SEND HTTP REDIRECT("/MyPage.HTM")

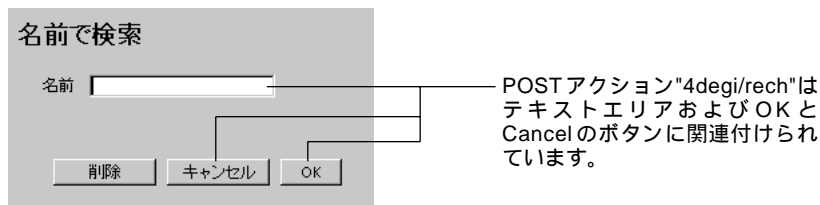
このコマンドがコンテキストモードで呼び出されると、実行された直後にWebプロセスはアボートされ、Webライセンスは開放されます。このコマンドは、同じメソッド内にあるデータを送るコマンド(**SEND HTML FILE**、**SEND HTML BLOB**など)より優先されます。

このコマンドは、他のWebサーバへ再問い合わせすることを可能にします。

4DはURLを自動的にエンコードしますが、*を渡すとエンコードしません。

例題

スタティックホームページを使って4D内で独自の検索を実行させるために、このコマンドを使用することができます。スタティックHTMLページ内に下記のエレメントを置いたと仮定します。



SET HTTP HEADER

SET HTTP HEADER(フィールド)

| 引数 | タイプ | 説明 |
|-------|-----|----------------|
| フィールド | 文字列 | HTTPヘッダ内のフィールド |

説明

このコマンドは、4DによってWebブラウザへ送り返されるHTTPヘッダ内のフィールドを設定できるようにするものです。これは、非コンテキストモードでのWebプロセスにのみ影響します。

このコマンドは、"cookies"の管理も可能にします。

HTTPヘッダ内のフィールドを設定したTextタイプのデータを引数フィールドへ渡します。

注：フィールドは常にcr/lfシーケンス（キャリッジリターン/ラインフィード）で分離されていなければなりません。構文についての詳細は、インターネットアドレス www.w3c.org で得られるR.F.Cs (Request For Comments)を参照してください。

通信プロトコルを指定しないと自動的に"HTTP/1.0 200 OK"が設定されます。

Charset、Expires、Content-Length、Content-Type、Date、Last-Modifiedのフィールドは、常に4Dによって設定されます。

例題

カスタム"cookie"の例を示します。

SET HTTP HEADER("SET-COOKIE:FUSER="+String(Abs(Random))+";PATH=/"

OPEN WEB URL

OPEN WEB URL(url{;.*})

| 引数 | タイプ | 説明 |
|-----|-----|---|
| url | 文字列 | スタートアップURL |
| * | | 指定されている場合=4DはURLを翻訳しない 省略されている場合=4DはURLを翻訳する |

説明

このコマンドは、Web ブラウザを起動し、url 引数で渡されたURLを開きます。

Web ブラウザが既に開かれている場合、このコマンドを実行すると、

Windows では、もうひとつのブラウザが起動し、url 引数で渡されたURLを開きます。

Macintosh では、現在のページに url 引数で渡されたURLを開きます。

このコマンドを実行したコンピュータ上から起動するブラウザがない場合は、何も行ないません。

4DはURLを自動的にエンコードしますが、*を渡すと4Dは翻訳しません。このオプションは、下記のタイプのURLにアクセスすることや、送ったりすることができます。

"http://www.server.net/page.htm?q=something"

注：このコマンドは、Web プロセスから呼び出された場合、何も行ないません。

例題

1. 下記のコードの行が実行されると、

OPEN WEB URL("file:///D:/web file.htm")

Web ブラウザが起動して、URL は下記の様に翻訳されます。

"file:///D%3A/web%20file.htm"

2. 下記のコードの行が実行されると、

OPEN WEB URL("file:///D:/web file.htm";*)

Web ブラウザが起動して、URL は"file:///D:/web file.htm"のままです。

3. 下記のコードの行はブラウザを起動して、ACI のホームページに接続します。

OPEN WEB URL("http://www.aci.co.jp/")

SET HOME PAGE

SET HOME PAGE(ホームページ)

| 引数 | タイプ | 説明 |
|--------|-----|---|
| ホームページ | 文字列 | ページ名またはHTML アクセスパス、またはカスタムホームページを送らないようにする空白文字列 |

説明

このコマンドを使用すると、現在のWeb プロセス用のカスタムホームページを定義することができます。デフォルトでは、コンテキストモードでメニューバー No.1 がホームページになります。

定義されたページはWeb プロセスに関連付けられているので、接続されたユーザごとに違うホームページを定義することもできます。このページはスタティックにもセミダイナミックにもなることができます。

HTML ホームページの名前またはページのHTML アクセスパスを、ホームページ引数に渡します。デフォルトホームページを有効にするには、空白をホームページに渡します。

注：バージョン6.5は、「データベースプロパティ」ダイアログ内で、デフォルトホームページを定義することができます（詳細は、第4章「Webサーバ」の「デフォルトホームページの定義」を参照してください）。この場合、Webサーバのスタートアップモード（コンテキストモードまたは非コンテキストモード）に関わらず、デフォルトホームページはすべてのWeb接続に適用されます。

ウインドウ

Open form window

Open form window({テーブル;}フォーム名{;タイプ{;水平位置{;垂直位置{;*}}}))

ウインドウ参照番号

| 引数 | タイプ | 説明 |
|-----------|------|---------------------------------|
| テーブル | テーブル | フォームのテーブル 省略されている場合デフォルトテーブル |
| フォーム名 | 文字列 | フォームの名前 |
| タイプ | 倍長整数 | ウインドウのタイプ |
| 水平位置 | 倍長整数 | ウインドウの水平位置 |
| 垂直位置 | 倍長整数 | ウインドウの垂直位置 |
| * | * | ウインドウの現在位置とサイズを保存 |
| ウインドウ参照番号 | 倍長整数 | ウインドウの参照番号 |

説明

この関数は、フォーム名引数で指定したフォームのサイズおよびサイズ変更プロパティを使用し、新しいウインドウを開きます。

フォームの内容はウインドウに表示されないことに注意してください。フォームを表示したい場合には、フォームをロードするコマンドを呼び出さなければなりません（例えば、**ADD RECORD**）。

デフォルト（タイプ引数が渡されていない場合）では、クローズボックス付きの標準ウインドウが開かれます。**Open window** コマンドとは異なり、ウインドウのクローズボックスには何のメソッドも定義されません。このクローズボックスをクリックすると、On Close Box フォームイベントがフォーム用に起動されている場合を除き、ウインドウをキャンセルして閉じます。この場合、On Close Box イベントに定義されたコードが実行されます。

フォームのサイズ変更が可能であれば、開かれたウインドウはズームボックスならびにグローボックスを持ちます。

注：フォームの主なプロパティを知るには、**GET FORM PROPERTIES** コマンドを使用します。

オプションのタイプ引数はウインドウのタイプが指定でき、下記の"Open window"テーマ内にある定数の内から1つを渡します。

| 定数 | タイプ | 値 |
|--------------------|------|-----|
| Plain window | 倍長整数 | 8 |
| Modal dialog box | 倍長整数 | 1 |
| Movable dialog box | 倍長整数 | 5 |
| Palette window | 倍長整数 | 720 |

オプションの引数<水平位置>は、ウインドウの水平位置を定義します。ポイント単位で位置を指定します（**Open window** コマンドを参照してください）。または、下記の"Window position"テーマ内にある定数の内から1つを渡します。

| 定数 | タイプ | 値 |
|-----------------------|------|--------|
| Horizontally Centered | 倍長整数 | 65536 |
| On the Left | 倍長整数 | 131072 |
| On the Right | 倍長整数 | 196608 |

オプションの引数垂直位置は、ウインドウの垂直位置を定義します。ポイント単位で位置を指定します（**Open window** コマンドを参照してください）。または、下記の"Window position"テーマ内にある定数の内から1つを渡します。

| 定数 | タイプ | 値 |
|---------------------|------|--------|
| Vertically Centered | 倍長整数 | 262144 |
| At the Top | 倍長整数 | 327680 |
| At the Bottom | 倍長整数 | 393216 |

これらの引数は、ツールバーおよびメニューバーの存在ならびにアプリケーションウインドウの現在のサイズ（Windowsの場合）を考慮に入れます。

オプションの引数*を渡すと、ウインドウをクローズした時の位置およびサイズが記憶されます。ウインドウが再度開かれる時に、その前の位置とサイズが優先されます。この場合、垂直位置と水平位置の引数は最初にウインドウが開かれる時にのみ使用されます。

例題

1. 下記のステートメントは標準のウインドウをクローズボックス付きで開き、自動的にそれを「入力」フォームと同じサイズになるように調整します。フォームは、サイズ変更可能なものとして定義されているので、ウインドウもグローボックスおよびズームボックスを持ちます。

```
$winRef:=Open form window([Table1];"Enter")
```

- 2 下記のステートメントは、画面の左上部にあるフローティングパレットを開くものです。このパレットは、開かれるたびに前回ユーザが閉じた時の位置に表示されます。

```
$winRef:=Open form window([Table1];"Tools";Palette window; On the Left;At the Top;*)
```

参照

Open window、GET FORMS PROPERTIES

GET FORM PROPERTIES

GET FORM PROPERTIES({テーブル;}フォーム名;幅;高さ{;ページ数{;幅変更{;高さ変更{;タイトル}}}))

| 引数 | タイプ | 説明 |
|-------|------|--------------------------------------|
| テーブル | テーブル | フォームが属しているテーブル 省略されている場合デフォルトテーブル |
| フォーム名 | 文字列 | フォームの名前 |
| 幅 | 倍長整数 | フォームの幅（ピクセル単位） |
| 高さ | 倍長整数 | フォームの高さ（ピクセル単位） |
| ページ数 | 倍長整数 | フォームのページ数 |
| 幅変更 | ブール | True= 幅サイズ変更可 False= ウィンドウ幅固定 |
| 高さ変更 | ブール | True= 高さサイズ変更可 False= ウィンドウ高さ固定 |
| タイトル | テキスト | フォームのウィンドウタイトル |

説明

GET FORM PROPERTIES コマンドは、フォーム引数<フォーム名>でのプロパティを返します。

幅および高さの引数は、フォームの幅と高さをピクセル単位で返します。これはフォームプロパティのデフォルトウィンドウサイズプロパティの値です。

デフォルトウィンドウサイズプロパティに自動サイズを指定している場合、幅と高さはフォーム内の全オブジェクトおよび全変数が表示可能となるように計算されます。

デフォルトウィンドウサイズプロパティに指定サイズを指定している場合、幅と高さはプロパティ内の幅と高さに入力されたものになります。

デフォルトウィンドウサイズプロパティにオブジェクトを指定している場合、幅と高さはこのオブジェクトの位置を元に計算されます。

タイトル引数には、フォームに定義されているウィンドウタイトルを返します。名前が定義されていないとタイトル引数には空白を返します。

幅変更と高さ変更引数には、フォームの幅と高さがサイズ変更可である（True）か、ウィンドウの幅とウィンドウの高さが固定に設定されている（False）かを、それぞれに返します。

引数<ページ数>は、0ページを除いたフォーム内のページ数を返します。

参照

Open form window

変更されたコマンド

この節では、バージョン6.5で構文が変更された既存のコマンドについて説明します。これらのコマンドはアルファベット順に紹介します。

新しい引数は、斜体で書かれています。

DISTINCT VALUES (配列コマンド)

DISTINCT VALUES(フィールド;配列)

| 引数 | タイプ | 説明 |
|-------|---------------------|-----------------------------------|
| フィールド | フィールド またはサブフィールド | 使用するインデックスが使えるフィールド またはサブフィールド |
| 配列 | 配列 | フィールドから値を受け取るための配列 |

説明

このコマンドは、下記の2つの点が変更されました。

DISTINCT VALUESは、すべてのインデックス可能なフィールド（インデックスが設定されている必要はないが、インデックスをサポートするタイプ）を指定できます。しかし、インデックスされていないフィールドでこのコマンドを実行すると、実行は遅くなります。

DISTINCT VALUESは、登録される前の新規レコードは考慮されません。

注：このコマンドはインデックスされたフィールドおよびインデックスされていないフィールドの両方に適応できます。実行モードを、**SET DATABASE PARAMETER** コマンドで設定することができます。

GOTO AREA(入力制御コマンド)

GOTO AREA({*; }オブジェクト)

| 引数 | タイプ | 説明 |
|--------|--------|---|
| * | | 指定している場合 = <オブジェクト> はオブジェクト名 (文字列) 省略している場合 = <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 (* が指定されている場合) または、フィールドか変数 (* が省略されている場合) |

説明

このコマンドの構文は、引数を選択できるように変更されました。

フィールドまたは変数の参照 (これまでと同じ)

例 : **GOTO AREA** ([Personnel]Name)

オブジェクト名での参照

例題

GOTO AREA (*;"MyVar")

注 : このコマンドは、入力フォームでのみ機能します。

Open document (システムドキュメントコマンド)

Open document(ドキュメント{;ファイルタイプ{;モード}}) ドキュメントファイル参照番号

| 引数 | タイプ | 説明 |
|---------|----------------|----------------|
| ドキュメント | 文字列 | ドキュメント名 |
| ファイルタイプ | 文字列 | ドキュメントタイプ |
| モード | 整数 | ドキュメントの開くモード |
| 戻り値 | ドキュメントファイル参照番号 | ドキュメントファイル参照番号 |

説明

オプションの引数<モード>が追加されました。これは、ドキュメントがどのように開かれるかを指定することができるようにするものです。

4つのオープンモードが指定可能です。4th Dimensionには下記の定数が定義されています。

| 定数 | タイプ | 値 |
|------------------------|-----|---|
| Read and Write(デフォルト値) | 整数 | 0 |
| Write Mode | 整数 | 1 |
| Read Mode | 整数 | 2 |
| Get Pathname | 整数 | 3 |

<モード>に3を渡すと!!00:00:00!!(ドキュメントの参照無し)を返します。ファイルは開かれませんが、documentとシステム変数OKは更新されます。

システム変数OKは1になります。

documentには、ドキュメントに渡された値に応じて、名前またはドキュメントのフルパスのいずれかがセットされます(ファイル名を渡すとdocumentはこの名前になります。フルパスを渡すとdocumentはこのフルパスになります)。

注: docNameに設定されたファイルが見つからない場合、またはドキュメントに空白を渡した場合は、「オープンファイル」ダイアログボックスが現れます。これが受け入れられると、Documentとシステム変数OKは上記のように更新されます。これが受け入れられないと、システム変数OKは0になります。

例題

1. Write Modeで既に関われているファイルをリードすることができます。

`vDoc:=Open document("PassFile";"TEXT")` `ファイルのオープン

2. ファイルが閉じられる前に、読み出し専用モードで検査することができます。

`vRef:=Open document("PassFile";"TEXT";Read Mode)`

PROCESS PROPERTIES (Processes)

PROCESS PROPERTIES(プロセス;名前;ステータス;時間{;表示{;ユニーク ID{;派生元}}})

| 引数 | タイプ | 説明 |
|---------|------|-------------------------|
| プロセス | 倍長整数 | プロセス番号 |
| 名前 | 文字列 | プロセス名 |
| ステータス | 整数 | プロセス状態 |
| 時間 | 整数 | 現在のプロセスの実行時間 |
| 表示 | ブール | 可視(True)または不可視 (False) |
| ユニーク ID | 整数 | 独自のプロセス番号 |
| 派生元 | 倍長整数 | プロセスの派生原因 |

説明

2つの新しいオプションの引数、 <ユニーク ID> および <派生元> がこのコマンドに追加されました。

<ユニーク ID> は、独自のプロセス番号を返します。

バージョン6.5からは、各プロセスはプロセス番号とセッションごとの独自のプロセス番号を持っています。独自のプロセス番号は、2つのプロセス間または2つのプロセスセッションの間で区別することができるようにするものです。これは、4th Dimension セッションの間に開始されたプロセス番号に対応するものです。

<派生元> は、プロセスの発生原因を表わす値を返します。

4th Dimension は、下記の定数を定義しています（Process Type テーマ内で）。

| 定数 | タイプ | 値 |
|------------------------------|------|-----|
| Web Process with License | 倍長整数 | -11 |
| Other 4D Process | 倍長整数 | -10 |
| External Task | 倍長整数 | -9 |
| Event Manager | 倍長整数 | -8 |
| Apple Event Manager | 倍長整数 | -7 |
| Serial Port Manager | 倍長整数 | -6 |
| Indexing Process | 倍長整数 | -5 |
| Cache Manager | 倍長整数 | -4 |
| Web Process with no License | 倍長整数 | -3 |
| Design Process | 倍長整数 | -2 |
| User or Custom Menus Process | 倍長整数 | -1 |
| None | 倍長整数 | 0 |
| Created from Programming | 倍長整数 | 1 |
| Created from Menu Command | 倍長整数 | 2 |
| Created from User Mode | 倍長整数 | 3 |
| Other User Process | 倍長整数 | 4 |

注：4D の内部プロセスは負の値を返し、ユーザが作成したプロセスは正の値を返します。

Semaphore (プロセス (通信) コマンド)

Semaphore(セマフォ{;待ち時間}) ブール

| 引数 | タイプ | 説明 |
|------|-----|--|
| セマフォ | 文字列 | テストし設定するセマフォ |
| 待ち時間 | 整数 | 最大の待ち時間 |
| 戻り値 | ブール | セマフォが正しく作成された(False)、またはセマフォは既に作成されている(True) |

説明

オプションの引数 <待ち時間> がこのコマンドに追加されました。この引数は、セマフォが既に設定されている場合、開放されるまでの待ち時間 (Tickで) を指定することができます。

この場合、関数はセマフォが開放されるか (Falseを返す) または待ち時間が切れる (Trueを返す) まで待ちます。

このオプションを使いたくない場合は、0を待ち時間に設定するか、またはこの引数を省略します。

例題

下記のコードを考えてみましょう。

```
t0:=Tickcount
onContinue:= True
While (Semaphore("flag") & onContinue)
    DELAY PROCES(Current process;10)
    If (Tickcount - t0 > 600)
        onContinue:= False
    End if
End while
If (Not(Semaphore("flag")))
    `何らかの処理...
End if

...は下記のように置き換えることができます。
If (Not(Semaphore("flag";600))) `10秒以上は待ちません
    `何らかの処理...
End if
```

SEND HTML FILE (Web サーバコマンド)

SEND HTML FILE(HTML ファイル)

| 引数 | タイプ | 説明 |
|-----------|-----|---|
| HTML ファイル | | HTML ファイルへの HTML アクセスパス、または SEND HTML FILE を終了させる空白文字列 |

説明

SEND HTML FILE コマンドは、システム変数 OK をセットするようになりました。送るファイルが存在していて、タイムアウトになっていなければ、OK は 1 になります。そうでなければ 0 になります。

SET INDEX (ストラクチャアクセスコマンド)

SET INDEX(フィールド;インデックス{;モード}{;*)

| 引数 | タイプ | 説明 |
|--------|-----------------|------------------------------------|
| フィールド | フィールドまたはサブフィールド | インデックスを作成または消去するフィールド |
| インデックス | ブール | インデックスを作成(True)またはインデックスを消去(False) |
| モード | 倍長整数 | インデックスモード (パーセントで) |
| * | | * が指定された場合は、非同期にインデックスを作成 |

説明

このコマンドの構文は、バージョン 6.5 の新しいインデックスモードの利点を取り入れられるように変更されました (詳細は、第 6 章「最適化」の「新しいインデックスモード」の節を参照してください)。

モードを指定すると、コマンドは新しいインデックスシステムを使用します。そうでない場合は、標準モードを使用します。

コマンドにインデックス作成を指定した場合 (インデックス引数が True の場合) にのみ引数 <モード> が使われます。

パーセンテージを表わす整数値を引数 <モード> に設定する必要があります。この値は、インデックスを最も有効に活用するタイプを指示することができます。これは下記の間でなければいけません。

mode=0 : インデックスはレコードの追加や挿入用に最適化されます。

mode=100 : インデックスは検索用に最適化されます。

例題

1. 下記のコードは、[Clients]Name フィールドのインデックスを作成します。このフィールドは主に検索に使用されます。

```
SET INDEX([Clients]Name;True;100)
```

2. [Prospects]Name フィールドのインデックスを作成します。このフィールドは、レコードの追加や挿入および検索をする時に使用されますが、追加や挿入の方が検索よりも頻繁に行なわれます。

```
SET INDEX([Prospects] Name;True;30)
```

SET WEB TIMEOUT (Web Server)

SET WEB TIMEOUT(タイムアウト)

説明

timeout の値は Web プロセスから呼び出された時、呼び出したプロセスにのみ適用され、他のプロセスには影響しません。

Web プロセス以外から呼び出された時は、以前のバージョンと同様に働きます。

フォームイベント

バージョン6.5には、3つの新しいフォームイベントが加えられました。

On After Keystroke

(更に、従来の On Keystroke イベントは On Before Keystroke に変更されました)

On Timer

On Resize

更に、フォームイベントの On Clicked は出力フォームでも発生します。

On After Keystroke

新しいフォームイベントである On After Keystroke は、エリア内に入力される文字をより良く管理し、フィルタすることを可能にしています。これは主に、新しいコマンド **Get edited text** と共に、また On Before Keystroke (On Keystroke の新しい名前、上記参照) と組み合わせて使用されることを想定しています。

このフォームイベントは、下記のように機能します。

文字がタイプされるとすぐに、イベント On Before Keystroke が発生します。このイベント内で **Get edited text** 関数を実行すると、最後のキーストロークの前の内容を返します。例えば、既にエリアに "PA" が入力されており、更に "R" をタイプすると On Before Keystroke イベントが発生し、そこで **Get edited text** 関数を実行すると "PA" を返します。ユーザがタイプを始める前にエリアに何もなければ、**Get edited text** 関数は空白を返します。

次に、新しいフォームイベント On After Keystroke が発生します。このイベント内で **Get edited text** 関数を実行すると、最後に入力された文字を含むエリアの内容を返します。

例えば、既にエリアに "PA" が入力されており、更に "R" をタイプすると、On After Keystroke イベント内での **Get edited text** 関数は "PAR" を返します。

これらの2つのイベントは、関連するオブジェクトメソッドでのみ使用できます。これらは、**POST KEY** コマンドによっても発生させることができます。

On Timer

この新しいフォームイベントは、フォームが画面上に表示されている間、**SET TIMER** コマンドによって定義されたTick数間隔で発生します。

On Timer フォームイベントを受け取るには、**SET TIMER** コマンドを呼び出す必要があります。このコマンドは、イベントの発生を起動し、発生間隔の時間 (Tick) を定義します (詳細は、**SET TIMER の説明**を参照してください)。

SET TIMER コマンドは、フォームメソッドまたはフォームのオブジェクトメソッドで呼び出さなければなりません。On Timer フォームイベントは、フォームメソッドでのみ使用できます。

注：4D Web サーバはこの機能を利用し、実行中のブラウザ上のHTMLページをダイナミックに直接更新させることができます。詳細は、**SET TIMER の説明**を参照してください。

On Resize

この新しいフォームイベントは、フォームウィンドウのサイズが変更された時に発生します。ウィンドウはいろいろな方法でサイズの変更が可能で、4Dがフォームオブジェクトのサイズを変更しなければならない時に、このイベントが発生することに注意してください。

したがって、On Resize フォームイベントは、下記の場合に発生します。

マウスを使ってウィンドウを拡大/縮小した場合：

ズームボックスの使用、ウィンドウ枠をドラッグ (Windows の場合)、最大化/最小化ボックスの使用など。

MINIMIZE WINDOW または **MAXIMIZE WINDOW** コマンドを呼び出した場合。

フォームを開く場合、出力モードから入力モードへの移動またはその逆、サブレコードを開く場合など。

このフォームイベントと **MOVE OBJECT / GET OBJECT RECT** コマンドの組合せは、フォーム内のオブジェクトの位置決めを完全に管理できるようになります。

On Clicked (変更点)

On Clicked フォームイベントは、**MODIFY SELECTION** または **DISPLAY SELECTION** コマンドによって表示される出力フォーム (リストモード) 内で、ユーザがフォームの本体をクリックすると発生します。

例えば、選択されたレコードに応じて、フッタまたはメニューアイテム内のボタンを有効、あるいは無効にすることが可能です (On Clicked フォームイベントが発生した時に UserSet セットを確認するだけです)。

注：ユーザモードで表示されるリストまたはサブフォームには適用されません。

On Double Clicked (変更点)

On Double Clicked フォームイベントは、**MODIFY SELECTION** または **DISPLAY SELECTION** コマンドによって表示される出力フォーム（リストモード）内で、ユーザがレコードをダブルクリックすると発生します。

このイベントは使用する入力フォームの変更を可能にします。また、**FILTER EVENT** コマンドを使用して、入力フォームへのアクセスを拒否することもできます。

注：ユーザモードで表示されるリストまたはサブフォームには適用されません。

デバッガ

バージョン6.5は、デバッガの機能と提供する情報に関しての変更や改善が実施されています。

追加の情報が表示されます。

ウインドウの管理が変更されました（複数のデバッガウインドウを同時に表示させることが可能です）。

新しい機能が付きました。

ウインドウのカレントパラメータの保存、コードの選択

Traceモードへの切り替えを行うキーボードコンビネーションが変更されました。

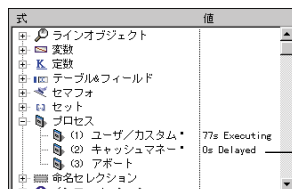
追加情報

デバッガでは、新しい情報が得られます。

注：デバッガウインドウ内で提供される情報のほとんどは、4Dのすべてのモードでのランタイムエクスプローラのヘルプによってアクセス可能です（第2章「デザインモード」の「ランタイムエクスプローラ」の節を参照してください）。

ロックされたレコードの数

「テーブル&フィールド」のリストは、各テーブルの上の方に「ロックされたレコード / [テーブル名]」と名付けられた、新しい行を持っています。これは、テーブル中の現在のロックされたレコード数を示すものです。



テーブル中のロックされたレコードの数

各プロセスによって使用された時間

プロセスリスト内では、各プロセスによって使用された時間が表示されます。

| プロセス | プロセスに割り当てられた時間 |
|--------------------|-------------------|
| (1) ユーザ/ 仮想マシンプロセス | 38s Executing |
| (2) キャッシュマネージャ | 0s Delayed |
| (3) デザインプロセス | 16s Waiting Event |

キャッシュの統計

この新しいリストは、4Dのキャッシュ内にロードされているテーブル、インデックスページおよび命名セクションの利用に関する統計を表示します。

ウインドウの管理

デバッグウインドウの管理が変更されました。バージョン6.5のデバッグウインドウは、アクティブウインドウと同じレベルに置かれます。他の多くのウインドウが表示されている時に、1つのウインドウから別のウインドウへ移動しても、デバッグウインドウを消してしまうということはありません。

結果として、トレースモードで異なるウインドウを同時に開くことが可能になりました。更に、デバッグを呼び出すウインドウ形式がモーダルである場合、デバッグウインドウもモーダルとなります。

新しい機能

デバッグは、メソッドの実行がどのように管理されるか、またその使いやすさを改善するための新しい機能を提供します。

ウインドウ構成の保存

新しいデバッグウインドウの表示には、同じセッション内で表示された最後のデバッグウインドウと同じ構成（ウインドウのサイズと位置、分割線の配置および「カスタム表現式/値」エリアの内容）を使用します。

現在のデバッグウインドウの構成を、データベースが開かれるたびにデフォルトでできるように保存することができます。この場合、デバッグのコントロールパネル内の「保存」ボタンをクリック、またはF3キーをクリックしなければなりません。

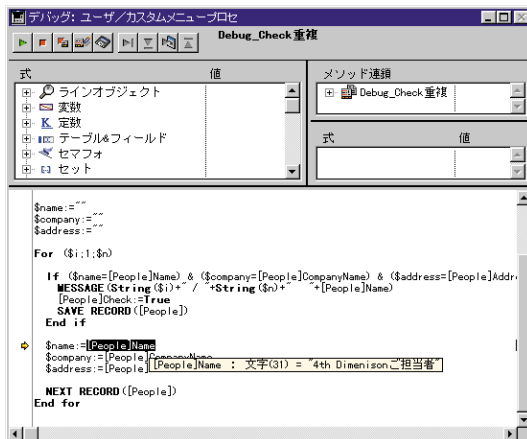


現在のウインドウ構成の保存ボタン

これらの内容は、データベースのストラクチャファイルに保存されます。

テキストの部分の選択および評価

ソースコードエリア内でテキストの部分を選択することが可能になりました。この場合、選択されたテキストの上にカーソルを移動すると、選択されたオブジェクトの値を Tips として表示します。



変数名またはフィールドをクリックすると自動的に選択されます。

選択されたテキストは、「カスタム表現式/値」エリア内に、3つの方法で置くことができます。

単純にドラッグ&ドロップを行なうことによって。選択されたテキストをクリックし、それを「カスタム表現式/値」エリア内へドラッグ&ドロップします。

Windows ではCtrl キーを、Macintosh ではcommandのキーを押しながら、選択されたテキストをクリックします。

「Ctrl + D」キー（Windows の場合）または「command + D」キー（Macintosh の場合）のコンビネーションを使ういます。

トレースモードへの切り替え

ユーザモードで、直接メソッドをトレースすることができるようになりました。メソッドの「手動」での中断手順も変更されました。

注：メソッドエディタ内でブレイクポイントを直接設定できるようになりました。詳細は、第2章「デザインモード」の「ブレイクポイントの挿入」を参照してください。

トレースモードでのメソッドの実行

ユーザモードで、直接メソッドをトレースすることができるようになりました。以前のバージョンでは、メソッドの始めにTRACE コマンドを挿入しなければなりませんでした。

これを行うには、メソッド名を選択した後に「メソッド実行」ダイアログボックスの「デバッグ」ボタンをクリックします（詳細は、この章の「メソッドの実行」の節を参照してください）。

4D Server：「特別」メニューの「メソッド実行...」から、「4D Server 上」オプションが選択されている場合には「デバッグ」ボタンは選択不可です。

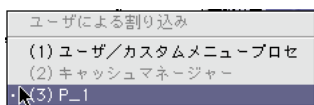
メソッドを中断するショートカットの変更

トレースを行なうために、メソッドの実行を中断させる方法が変わりました。以前のバージョンでは、Altキーを押しながらクリック（Windowsの場合）またはoptionキーを押しながらクリック（Macintoshの場合）していましたが、バージョン6.5では下記のコンビネーションになりました。

Windowsでは、「Alt + Shift」キーを押しながら右クリックします。

Macintoshでは、「Control + option + command」キーを押しながらクリックします。

この動作を実行すると、実行されているすべてのプロセスをリストしたポップアップメニューが現れ、トレースしたいプロセスを選択することができます。



ユーザがクリックしたウインドウ内のプロセス名の隣にマークが置かれ、トレースすることのできないプロセスはグレーで表示されます。

注：以前のバージョンでは、**ON ERR CALL**が設定されて、Altキーを押しながらクリック（またはoptionキーを押しながらクリック）によってメソッドが中断されると、エラー1066が4DのErrorシステム変数に返されましたが、このエラーは発生しなくなりました。

他の章で述べられた、バージョン6.5に加えられた新しい機能の他に、このバージョンでは多くの「内部的」最適化も取り入れています。これらの最適化は4Dデータベースの性能を改善するのものです。

新しいインデックスモード

4Dは、2つのインデックスモードを選択できます。ひとつは「従来」モードであり、以前のバージョンで使用されてきた方法です。もうひとつは、新しい「高速」モードで、ほとんどの場合に速度を上げて処理することができます。

インデックスモードの選択

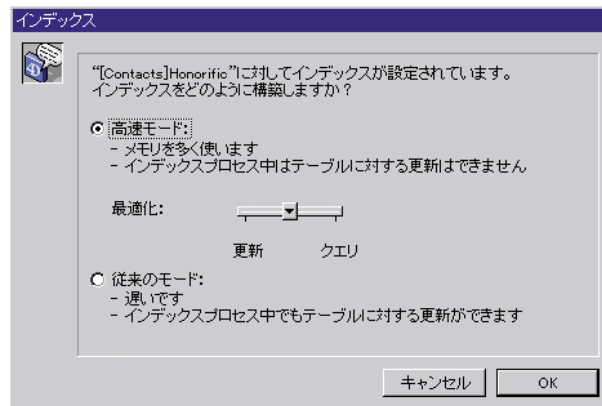
インデックスモードの選択は、レコードの数が1000を超える場合にのみ、デザインモードで変更可能です。

インデックスモードを選択するには、

- 1 ストラクチャエディタへ移る。
- 2 インデックスあるいは再インデックスするフィールドを選び、ストラクチャメニュー内のフィールドプロパティを選択するか、インデックスあるいは再インデックスするフィールドをクリックする。
- 3 フィールドプロパティのパレット内で、インデックスオプションを選択し、「適用」ボタンをクリックする。

注：フィールド名は、マウスの右ボタンをクリック（Winodwsの場合）またはControlキーを押しながらクリック（Macintoshの場合）し、コンテキストメニュー内のインデックスあるいは再インデックスを選択することもできます。

下記のダイアログボックスが現れます。



- 4 使用したいモードに対応したラジオボタンをクリックする。
どのモードを選ぶかは、次項を参照してください。
- 5 高速モードを選んだ場合、サーモメータを使用してインデックスの最適化率を設定する。

この率は、インデックスがどのように使用されるかによって設定されなければならない、且つ、下記の2つの限度の範囲内であればなりません。

更新：インデックスは一定して変更されます。これは、データが一定してテーブルに加えられる場合です。

クエリ：インデックスは変更されず、主に検索やソートに使用される場合です。

これは、インデックスの最適化率に関わるもので、その定義に関わるものではありません。サーモメータを右端のクエリ側に指定したとしても、インデックスはまだ変更することができます。しかし、その能力は最適な状態にはなりません。

どちらのモードを選ぶか？

それぞれのモードには長所も短所もあります。

2つのインデックスモードの比較表

| | 長所 | 短所 |
|-------|---|--|
| 高速モード | <ul style="list-style-type: none">- インデックスの構築が速い¹- 用法に応じた最適化の可能性 | <ul style="list-style-type: none">- メモリーをより多く使用する- インデックスプロセスの間はテーブルへの変更はできない |
| 従来モード | <ul style="list-style-type: none">- メモリーの必要量が少ない- インデックスプロセスの間でも、テーブルの変更が可能である | <ul style="list-style-type: none">- インデックスの構築が遅い- 最適化が行えない |

1. 速度の増加は、主にインデックスするレコード数、データのフラグメンテーションの度合いおよびメモリ（RAM）の容量によります。

まとめると、高速モード（デフォルトで選択）は、より効率的で最適化することができます。ほとんどの場合、こちらを使うようにしてください。速度の最大限の増加を図るためには、インデックス動作をメモリ内なるべく頻繁に行なうようにしなければなりません（ランタイムエクスプローラのウォッチページ内で、メモリーキャッシュがインデックスを保持するに十分であるかどうかを確認することができます）。

しかし、メモリが足りない場合、または一定時間の間はデータベース内のテーブルをロック（書き込みモードで）させたくない場合には、従来モードを選ぶことができます。

データベース機能の速度増進

4Dのデータベース機能の多くは最適化されています。すなわち、これらの実行速度は飛躍的に速くなったことを意味します。変更された機能を下記に紹介します。

検索およびシーケンシャルソート

検索およびシーケンシャルソートは最適化されました。4Dはディスクからより多くの記録を一度に読み出そうとします。キャッシュのサイズが小さい時、このモードは特に興味深いものです。

更に、これらの関数を呼び出す機能の性能も改善されました。

書き出し用コマンド

読み込み用コマンド

SELECTION TO ARRAY コマンド

注：**SET DATABASE PARAMETER** ルーチンを使って、最適化モードを無効にすることもできます。

インデックス検索

インデックス検索は、データが多くの繰り返された値で成り立っている場合に最適化されます。

サーモメータの表示開始の遅れ

特定の処理を実行している時に、サーモメータの表示開始が遅れます。下記の関数で処理に時間がかかる場合（1秒以上）に、サーモメータの表示を遅らせて現れるようにします。以前のバージョンでは、実行時間が非常に短い場合には、検索よりもサーモメータの表示により多くの時間がかかっていました。

注：この機能は、すでに検索およびインデックスソートに適用されています。

次のコマンドが影響を受けます：

DISTINCT VALUES、**RELATE ONE SELECTION**、**RELATE MANY SELECTION**、**ORDER BY**、**QUERY**、ならびにインデックスを消去する関数。

インデックスページをコンパクト化

多くのインデックスやレコードを持つデータベースのインデックスページは、4Dのメモリキャッシュを多く使います。

キャッシュがいっぱいで4Dが追加のスペースを必要とする場合には、キャッシュ内のデータが直接アンロードされます。この操作を実行する前に、プログラムは、インデックスページをコンパクト化することによってスペースを得られるかどうかをチェックしようとしています。この選択肢により、データを後で再ロードするのを防ぐことができます。

注：**SET DATABASE PARAMETER** ルーチンを使って、この機能を無効にすることもできます。

B

BOOLEAN ARRAY FROM SET(ブール配列{;セット})120

C

CREATE SELECTION FROM ARRAY
({テーブル;}レコード配列{;セクション名})130

CREATE SET FROM ARRAY({テーブル;}レコード配列{;セット名})151

D

DISTINCT VALUES(フィールド;配列)172

E

EXECUTE ON CLIENT(クライアント名;メソッド名{;引数 1;...;引数 N})143

EXPORT DATA(ファイル名{;プロジェクト{;*}})128

F

Find index key(インデックスフィールド;値) 倍長整数147

G

Get database parameter({テーブル;}セクタ) 倍長整数154

Get edited text Text124

GET FORM PROPERTIES({テーブル;}フォーム名;幅;高さ
{;ページ数{;幅変更{;高さ変更{;タイトル}}}))171

GET OBJECT RECT({*;}オブジェクト;左;上;右;下)131

GET REGISTERED CLIENTS(クライアントリスト;メソッド)142

GOTO AREA({*;}オブジェクト)173

H

HIGHLIGHT RECORDS({セット名})150

I

| | |
|--------------------------------------|-----|
| IMPORT DATA(ファイル名{;プロジェクト{*}}) | 126 |
| Is new record({テーブル}) ブール | 148 |
| Is record loaded({table}) ブール | 149 |

L

| | |
|--|-----|
| LOG EVENT(メッセージ{;重要度}) | 157 |
| LONGINT ARRAY FROM SELECTION ({テーブル;}レコード番号配列{;セクション名}) | 121 |

M

| | |
|---|-----|
| MOVE OBJECT(*;オブジェクト;水平移動;垂直移動; {;水平リサイズ{;垂直リサイズ}}{*}) | 133 |
|---|-----|

O

| | |
|--|-----|
| Open document(ドキュメント{;ファイルタイプ{;モード}} ドキュメントファイル参照番号 | 174 |
| Open form window({テーブル;}フォーム名{;タイプ{;水平位置 {;垂直位置{*}}}) ウィンドウ参照番号 | 168 |
| OPEN WEB URL(url{*}) | 166 |

P

| | |
|--|-----|
| PICT TO GIF(ピクチャ;GIF用BLOB) | 135 |
| Process aborted ブール | 145 |
| PROCESS PROPERTIES(プロセス;名前;ステータス;時間 {;表示{;ユニークID{;派生元}}}) | 175 |

Q

| | |
|--|-----|
| QUERY WITH ARRAY(インデックスフィールド;配列) | 146 |
|--|-----|

R

| | |
|--|-----|
| REGISTER CLIENT(クライアント名{;ピリオド}{*}) | 138 |
|--|-----|

S

| | |
|--|-----|
| Select folder({メッセージ}) 文字列 | 155 |
| Semaphore(セマフォ{;待ち時間}) ブール | 177 |
| SEND HTML BLOB(blob;タイプ{;非コンテキスト}) | 158 |
| SEND HTML FILE(HTML ファイル) | 178 |

| | |
|--|-----|
| SEND HTTP REDIRECT(url{,*}) | 163 |
| SET DATABASE PARAMETER({テーブル;}セクタ;値) | 152 |
| SET HOME PAGE(ホームページ) | 167 |
| SET HTTP HEADER(フィールド) | 165 |
| SET INDEX(フィールド;インデックス{;モード}{,*}) | 178 |
| SET TIMER(チック数) | 122 |
| SET WEB TIMEOUT(タイムアウト) | 179 |

T

| | |
|--------------------------------|-----|
| Test semaphore(セマフォ) ブール | 137 |
|--------------------------------|-----|

U

| | |
|-------------------------|-----|
| UNREGISTER CLIENT | 141 |
|-------------------------|-----|

W

| | |
|--|-----|
| WEB CACHE STATISTICS(ページ;ヒット数;使用率) | 162 |
| Web Context ブール | 161 |

記号

| | |
|------------------------|-----|
| .4SI | 81 |
| .shtm | 111 |
| /4DCASHECLEAR | 102 |
| /4DCGI | 111 |
| /4DHTMLSTATS | 102 |
| /4DSTATS | 101 |
| <?[テーブル名]> | 36 |
| <?[テーブル名]フィールド名> | 36 |

数字

| | |
|-------------------------------|-----|
| 0 ページ上のオブジェクト | 104 |
| 4D | 10 |
| 4D 3.x.x データベース | 10 |
| 4D 6.0.x データベース | 10 |
| 4D Help65J.RSR ファイル | 58 |
| 4D Help65J ファイル | 58 |
| 4D Server ネットワークコンポーネント | 10 |
| 4DACTION | 112 |
| Form を POST する | 114 |
| 4DMSG.DLL | 73 |
| 4DVAR MAVAR | 97 |

A

| | |
|--------------------|----|
| ASCII 変換テーブル | 98 |
|--------------------|----|

B

| | |
|------------------------------|-----|
| BOOLEAN ARRAY FROM SET | 120 |
|------------------------------|-----|

C

| | |
|-----------------------------------|-----|
| COMPILER_WEB | 115 |
| CREATE SELECTION FROM ARRAY | 130 |
| CREATE SET FROM ARRAY | 151 |
| Customizer Plus | 10 |

D

| | |
|-----------------------|-----|
| DISTINCT VALUES | 172 |
|-----------------------|-----|

E

| | |
|-------------------------|-----|
| EXECUTE ON CLIENT | 143 |
| EXPORT DATA | 128 |

F

| | |
|----------------------|-----|
| Find index key | 147 |
|----------------------|-----|

G

| | |
|------------------------------|-----|
| Get database parameter | 154 |
| Get edited text Text | 124 |
| GET FORM PROPERTIES | 171 |
| GET OBJECT RECT | 131 |
| GET REGISTERED CLIENTS | 142 |
| GOTO AREA | 173 |

H

| | |
|-------------------------|-----|
| HIGHLIGHT RECORDS | 150 |
| HTML サポート | 104 |
| HTTP リクエスト用 | 100 |

I

| | |
|------------------------|-----|
| IMPORT DATA | 126 |
| IP アドレス | 100 |
| ブラウザが動作しているマシンの | 87 |
| Is new record | 148 |
| Is record loaded | 149 |

L

| | |
|------------------------------------|-----|
| LOG EVENT | 157 |
| LONGINT ARRAY FROM SELECTION | 121 |

M

| | |
|-------------------|-----|
| MOVE OBJECT | 133 |
|-------------------|-----|

O

| | |
|---------------------------|-----|
| On After Keystroke | 180 |
| On Before Keystroke | 180 |
| On Clicked (変更点) | 181 |

| | |
|----------------------------------|-----|
| On Double Clicked (変更点) | 182 |
| On Keystroke (変更) | 180 |
| On Resize | 181 |
| On Timer | 181 |
| On Web Authentication データベースメソッド | 87 |
| On Web Connection データベースメソッド | 116 |
| Open document | 174 |
| Open form window | 168 |
| OPEN WEB URL | 166 |
| P | |
| PICT TO GIF | 135 |
| Process aborted | 145 |
| PROCESS PROPERTIES | 175 |
| Q | |
| QUERY WITH ARRAY | 146 |
| R | |
| REGISTER CLIENT | 138 |
| S | |
| Select folder | 155 |
| Semaphore ブール | 177 |
| SEND HTML BLOB | 109 |
| SEND HTML BLOB | 158 |
| SEND HTML FILE | 178 |
| SEND HTTP REDIRECT | 163 |
| SET DATABASE PARAMETER | 152 |
| SET HOME PAGE | 167 |
| SET HTTP HEADER | 165 |
| SET INDEX | 178 |
| SET TIMER | 122 |
| SET WEB TIMEOUT | 179 |
| T | |
| Test semaphore | 137 |
| Tips | |
| ダイナミックリファレンス | 37 |
| ピクチャとして表示されるボタン | 106 |
| U | |
| UNREGISTER CLIENT | 141 |
| W | |
| WEB CACHE STATISTICS | 162 |
| Web Context | 161 |

| | |
|------------------------------|-----|
| Web サーバ | 158 |
| .shtm | 112 |
| /4DCASHECLEAR | 101 |
| /4DHTMLSTATS | 101 |
| /4DSTATS | 101 |
| 4DACTION を POST する Form | 114 |
| 4D Web サーバのアクセスシステム | 89 |
| 4D 内の変換文字セットの変更 | 98 |
| HTML サポート | 104 |
| On Web Authentication | |
| 一般 Web ユーザ | 92 |
| キャッシュのクリア | 100 |
| セミダイナミックページ | 111 |
| スタートアップ時に | |
| 非コンテキストモードを定義する | 110 |
| スタティックホームページ用キャッシュ | 99 |
| 接続ログファイル | 103 |
| データ入力管理に JavaScript を使用 | 97 |
| データベースメソッド | 87 |
| デフォルト HTML ルートフォルダを | |
| 定義する | 93 |
| デフォルトホームページの定義 | 95 |
| パスワードオプション | 86 |
| 非コンテキストモード | 107 |
| WindowsNT の「イベントビューワ」 | 73 |
| あ | |
| 新しいコマンド | 120 |
| BOOLEAN ARRAY FROM SET | 120 |
| CREATE SELECTION FROM ARRAY | 130 |
| CREATE SET FROM ARRAY | 151 |
| EXECUTE ON CLIENT | 143 |
| EXPORT DATA | 128 |
| Find index key | 147 |
| Get database parameter | 154 |
| Get edited text Text | 124 |
| GET FORM PROPERTIES | 171 |
| GET OBJECT RECT | 131 |
| GET REGISTERED CLIENTS | 142 |
| GOTO AREA | 173 |
| HIGHLIGHT RECORDS | 150 |
| IMPORT DATA | 126 |
| Is new record | 148 |
| Is record loaded | 149 |
| LOG EVENT | 157 |
| LONGINT ARRAY FROM SELECTION | 121 |

| | | | |
|-----------------------------|-----|-------------------------|-----|
| MOVE OBJECT | 133 | オブジェクトプロパティ..... | 131 |
| Open document | 174 | オンラインヘルプ..... | 41 |
| Open form window | 168 | オンラインヘルプの作成..... | 42 |
| OPEN WEB URL | 166 | 4D データベースからの呼び出し..... | 43 |
| PICT TO GIF | 135 | か | |
| Process aborted | 145 | 階層リスト..... | 37 |
| PROCESS PROPERTIES | 175 | 拡張ASCIIコード..... | 98 |
| QUERY WITH ARRAY | 146 | 拡張文字を直接送信..... | 98 |
| REGISTER CLIENT..... | 138 | 各プロセスによって使用された時間..... | 184 |
| Select folder..... | 155 | カラーラベル..... | 38 |
| Semaphore | 177 | き | |
| SEND HTML BLOB | 158 | キーボード | |
| SEND HTML FILE | 178 | ストラクチャエディタ内..... | 15 |
| SEND HTTP REDIRECT..... | 163 | プロパティリスト..... | 31 |
| SET DATABASE PARAMETER..... | 152 | メソッドエディタ..... | 59 |
| SET HOME PAGE | 167 | キャッシュの統計..... | 184 |
| SET HTTP HEADER | 165 | キャッチページ..... | 47 |
| SET INDEX | 178 | く | |
| SET TIMER | 122 | クエリ..... | 146 |
| SET WEB TIMEOUT | 179 | グリッドにオブジェクトを作成..... | 29 |
| Test semaphore | 137 | け | |
| UNREGISTER CLIENT..... | 141 | 検索するオブジェクトタイプ..... | 49 |
| WEB CACHE STATISTICS | 162 | 検索のオプション..... | 50 |
| Web Context | 161 | こ | |
| 新しいファイルフォーマット..... | 80 | 互換性..... | 10 |
| い | | コマンドのシンタックスをプレ入力する..... | 57 |
| 一般Webユーザ..... | 92 | コメント..... | 51 |
| イベントビューワ..... | 73 | コメント用フォント属性の変更..... | 52 |
| う | | 自動コメントの挿入..... | 54 |
| ウインドウの管理..... | 184 | 日付、時間またはユーザ名の挿入..... | 53 |
| ウォッチページ..... | 45 | コンテキスト..... | 97 |
| え | | コンテキストメニュー | |
| エクスプローラ..... | 51 | ストラクチャウインドウ..... | 14 |
| エレメントを表示/非表示 | 19 | フォーム..... | 18 |
| お | | コンテキストモードから | |
| 大文字小文字を区別..... | 50 | 非コンテキストモードへの切り替え..... | 108 |
| オブジェクトタイプ | | し | |
| 検索..... | 49 | システムドキュメント..... | 155 |
| 変更..... | 32 | システム環境..... | 157 |
| オブジェクト検索エディタ..... | 47 | 自動スプリッタ..... | 35 |
| オブジェクト選択ショートカット..... | 18 | | |
| オブジェクトタイプと検索のスコープ..... | 49 | | |
| オブジェクトにコメントを付ける..... | 51 | | |

| | | | |
|----------------------|-----|-------------------------|-----|
| 出力フォーム | 40 | テキストの部分の選択および評価 | 185 |
| アクティブオブジェクト | 40 | デザインモード | 11 |
| 使用されているライセンスの数 | 110 | データ書き出し | |
| ショートカット | | 新しいエディタ | 76 |
| コンテキストメニュー | 18 | 新しいオプション | 78 |
| テキスト編集を操作する | 59 | 新しいファイルフォーマット | 80 |
| ピクチャボタン | 73 | データ入力管理に JavaScript を使用 | 96 |
| プロパティリスト | 31 | データファイルのパス | 13 |
| ポップアップメニュー | 73 | データファイルを作成 | 14 |
| メソッドエディタで | | データファイルを選択 | 14 |
| 新規データベースの作成 | 12 | データベース | 13 |
| シンタックスエラーを見る | 58 | データベースのオープン | 12 |
| シンタックスヘルプ | 56 | データベースプロパティ | |
| す | | スタートアップ時に | |
| スタートアップ時にクライアントを登録する | 63 | クライアントを登録する | 63 |
| スタティックホームページ | | デフォルトフォームの作成 | 17 |
| 4D 変数 | 97 | デフォルトユーザ | 65 |
| スタティックホームページ用キャッシュ | 99 | テンポラリフォルダ | 62 |
| ストラクチャアクセス | 152 | メソッドエディタ内でのフォントの定義 | 59 |
| ストラクチャウインドウ | | 文字「@」の管理 | 61 |
| インタフェース | 14 | ユーザモードへのグループのアクセス | 64 |
| テーブルの並びの管理 | 15 | データベースメソッド | |
| ストラクチャファイルをチェックする | 14 | On Web Authentication | 87 |
| スプリッタ | 33 | On Web Connection | 116 |
| 自動スプリッタ | 35 | データベース内の検索 | 47 |
| 隣接するオブジェクトプロパティとの | | データベースを作成 | 12 |
| 相互作用 | 34 | データベースを開く | 13 |
| せ | | データ読み込み | |
| 接続ログファイル | 102 | 新しいエディタ | 76 |
| セット | 151 | 新しいオプション | 78 |
| セマフォ | 119 | 新しいファイルフォーマット | 80 |
| セレクション | 150 | データ読み込みとデータ書き出し | 75 |
| た | | デバッグ | 183 |
| ダイナミックテーブル名または | | デバッグ | |
| ダイナミックフィールド名の挿入 | 36 | トレースモードでのメソッドの実行 | 186 |
| タブコントロール | 106 | メソッドの実行 | 83 |
| つ | | デバッグウインドウ | 184 |
| ツールパレット | 20 | デフォルトHTMLルートフォルダを定義する | 93 |
| て | | デフォルトフォームの作成 | 16 |
| テーブル | 15 | デフォルトフォームの自動作成 | 17 |
| テーブルの並び | 15 | デフォルトホームページの定義 | 95 |
| テーブル名 | 36 | デフォルトユーザ | 65 |
| | | テンポラリフォルダ | 62 |
| | | と | |
| | | トレースモードでのメソッドの実行 | 185 |

- ドラッグ&ドロップ 37
- トレースモードへの切り替え 185
- ドロップダウンメニュー 39
- ドロップダウンリスト 39
- は
- 背景ピクチャ 105
- 配列 120
- はじめに 9
- バックグラウンド 38
- パスワード 64
 - デフォルトユーザ 65
- パスワードオプション 86
- パスワード使用 86
- パスワードの作成と変更 66
- パレットのフォーカス自動管理 23
- ひ
- ピクチャ 135
 - 作成と変更 69
 - 追加 67
 - プロパティ 68
 - 見る 68
- ピクチャ変数 104
- ピクチャボタン 39
- ピクチャライブラリ 66
 - ショートカット 69
 - ピクチャの作成と変更 69
 - ピクチャの追加 67
 - フレームアシスタント 70
- 非コンテキストモード 107
 - コンテキストモードからの切り替え 108
- 非コンテキストモードから
 - コンテキストモードへの切り替え 109
- 非コンテキストモードで開始 110
- 表示
 - コメント 51
 - ツールパレット 20
 - マーカー 21
 - マーカーラベル 21
- ふ
- フィールド 37
- フィールド名 36
- フィールドを追加コマンド 18
- フィールドを連結 82
- フォーカス表示属性 39
- フォーム 16
 - デフォルトフォームの作成 16
 - フォームエディタの変更 17
- フォームイベント 180
 - On After Keystroke 180
 - On Before Keystroke 180
 - On Clicked (変更点) 181
 - On Double Clicked (変更点) 182
 - On Keystroke 180
 - On Resize 181
 - On Timer 181
- フォームウィザード 37
- フォームエディタ
 - エレメント表示/非表示 19
 - オブジェクトを複製 28
 - コンテキストメニュー 18
 - フォームの境界表示 22
 - フォームタイプの選択 26
 - フォームメソッドへの直接アクセス 23
 - プロパティリストの表示 29
- フォームエディタ内の
 - エレメント表示/非表示 19
- フォームオブジェクト 33
 - スプリッタ 33
- フォームタイプ 26
- フォームの境界表示 22
- フォームプロパティ
 - フォームタイプの選択 26
 - プロパティリスト内に表示 31
- フォームメソッド 23
- フォームメソッドへの直接アクセス 23
- フォントを定義 59
- プラグインエリア 104
- ブラケットの代りに
 - 4DVAR コメントを使用する 97
- ブラチナ調 38
- プラットフォームインタフェース 38
- ブレイクページ 47
- ブレイクポイント 60
- ブレイクリスト 47
- フレームアシスタント 70
- フレームの挿入と削除 71
- フレームの並び 70
- プロセス (通信) 137

| | | |
|---------------------|-----|--|
| プロパティリスト | | |
| ショートカット | 29 | |
| 表示 | 29 | |
| フォームプロパティを表示 | 31 | |
| フォーム内の変数のタイプ分け | 32 | |
| へ | | |
| ページ管理コマンド | 26 | |
| 変更されたコマンド | 172 | |
| 変数 | | |
| Tips 内のダイナミックリファレンス | 37 | |
| ウォッチページ | 45 | |
| フォーム内の変数のタイプ分け | 32 | |
| 変数の入力不可タイプ | 25 | |
| ほ | | |
| ポップアップ/ドロップダウンリスト | 39 | |
| ポップアップメニュー | 39 | |
| ま | | |
| マーカー | 21 | |
| マーカーラベル | 21 | |
| マトリックス上でオブジェクトを複製 | 28 | |
| め | | |
| 命名セクション | 130 | |
| メソッド | 57 | |
| メソッドエディタ | | |
| シンタックスを見る | 56 | |
| フォントの定義 | 59 | |
| ブレークポイントの挿入、変更、削除 | 60 | |
| メソッドエディタ内でのフォントの定義 | 59 | |
| メソッドの実行 | 83 | |
| メソッドの編集 | 56 | |
| メソッドを除外 | 50 | |
| メソッドを中断する | | |
| ショートカットの変更 | 186 | |
| メニューバーをフォームに関連付ける | 19 | |
| も | | |
| 文字セット | 98 | |
| 文字列フィールド | 16 | |
| ゆ | | |
| ユーザモード | 75 | |
| トレースモードへの切り替え | 185 | |
| よ | | |
| 用紙 | 20 | |
| ら | | |
| ラベル | 38 | |
| ラベルエディタ | 82 | |
| ランゲージ | 119 | |
| ランタイムエクスプローラ | 44 | |
| ウィンドウの表示 | 44 | |
| り | | |
| リスト項目 | 119 | |
| リストのドラッグ&ドロップ | 37 | |
| れ | | |
| レコード | 148 | |
| ろ | | |
| ロックされたレコードの数 | 183 | |
| ロボット | 92 | |
| ロールオーバー | 40 | |