



このドキュメントでは、4th Dimensionバージョン2004.4の新機能および変更点の概要が説明されています。

- **フォーミュラで利用できる4Dコマンド**：フォーミュラで利用できる4Dコマンドのアクセス制限を設定するための新しいオプションができました。
- **スペルチェック辞書の事前単語登録**：4th Dimension2004.4では、スペルチェックが認識する単語を事前に登録できるようになりました。
- **バックアップおよび復元**：4th Dimensionの統合されたバックアップモジュールの使い勝手が向上しました。
- **Right-to-left言語のサポート**：右から左に記述する言語のアプリケーション開発を支援するため、デザインモードおよび4th Dimensionのランゲージに変更が加えられました。
- **アプリケーションビルド**：「4Dサーバへの接続」ダイアログの表示を強制できるようになりました。また、データファイルの場所を特定するための新しいXMLキーが設けられました。
- **ランゲージ**：**Get form parameter** コマンドが追加され、**SET DATABASE PARAMETER**、**Get database parameter**、**DOM SET XML ELEMENT VALUE** および **SET ABOUT** コマンドに変更が加えられました。
- **4D View**：4D View 2004.4では、スクロールおよび選択の解除がコマンドで実行できるようになりました。
- **4D Pack**：**AP Add table and fields** コマンドが詳細なエラーコードを返すようになりました。

フォーミュラで利用できる4Dコマンド

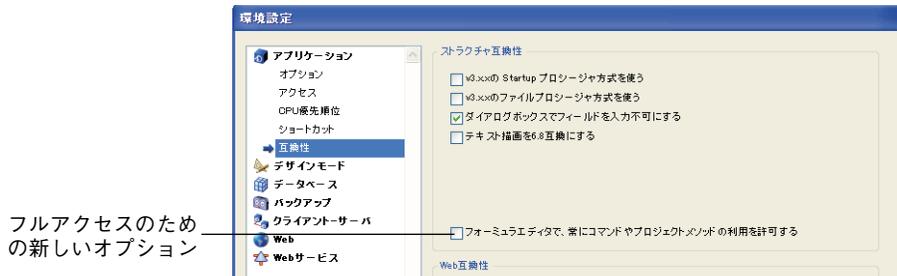
4th Dimensionバージョン2004では、セキュリティ上の理由により、標準エディタ群（フォーミュラエディタ、クイックレポートエディタなど）には新しい基本原則が適用され、特定のコマンド、または**SET ALLOWED METHODS** コマンドで明示的に許可されたプロジェクトメソッド以外は実行できないようになりました。

変換されたデータベースの場合、バージョン 2004 の新しい基本原則によって操作性が著しく低下したように感じるかもしれません。一部のデベロッパからは、以前の 4th Dimension との互換性（標準エディタでは、すべてのコマンドおよびプロジェクトメソッドが実行できる）を求める声が上がっていました。

こうした要望に応え、4th Dimension 2004.4 では環境設定に新しい項目が 2 つ設けられました。

フルアクセス（互換モード）

4th Dimension 環境設定の互換性ページ（「アプリケーション」テーマ）には、「フォーミュラエディタで、常にコマンドやプロジェクトメソッドの利用を可能にする」という新しいオプションが設けられました。



このオプションが有効にされている場合、4th Dimension は以前のバージョンと同じように動作し、標準のエディタを使用してプラグインコマンドを含むすべてのコマンドおよびプロジェクトメソッド（プロパティで隠されたメソッドを除く）が実行できるようになります。実行できるのは、フォーミュラエディタのコマンドリストに表示されるコマンドおよびメソッドです。

注：このオプションは SET ALLOWED METHODS コマンドよりも優先度が高いため、有効にされている場合、同コマンドによる設定が無効になります。

バージョン 2004 よりも前のバージョンから変換されたデータベースの場合、このオプションはデフォルトで有効になっています。それ以外の場合、このオプションはデフォルトで無効になっています。

Designer と Administrator にはフルアクセスを許可する

4th Dimension 2004.4 では、Designer と Administrator には 4D コマンドおよびプロジェクトメソッドに対するフルアクセスを許可するためのオプションが追加されました。

Designer と Administrator に限定されたフルアクセスは、4th Dimension 環境設定のアクセスページ（「アプリケーション」テーマ）にある「Designer と Administrator に対する、フォーミュラエディタ上での、コマンドやプロジェクトメソッドのフィルタを無効にする」というオプションを使用して設定します。

限定的なフルアクセスのための新しいオプション



限定的なフルアクセスの利点は、すべてのコマンドおよびプロジェクトメソッドが実行できると同時に、その操作を責任あるユーザに限定することによって管理ができています。

開発中の段階では、このモードを使用することにより、あらゆるフォーミュラやレポートを自由にテストすることができます。

運用中の環境では、一時的にすべてのコマンドが実行できるようにするための手段としてこのオプションを利用することができます。つまり、ログインユーザを変更し（**CHANGE CURRENT USER** コマンド）、フルアクセスが許可された状態でダイアログまたはレポート出力プロセスをコールし、特定の操作が終了した時点でログインユーザを元に戻すことにより、目的の操作を実行することができます。

注：前述の「フォーミュラエディタで、常に...」オプションによってフルアクセスの互換性モードが設定されている場合、限定的なフルアクセスのオプションは無効になります。

4th Dimension バージョン 2004 以降で作成されたデータベースの場合、このオプションはデフォルトで有効になっています。それ以外の場合、このオプションはデフォルトで無効になっています。

スペルチェック辞書の事前単語登録

4th Dimension 2004.4 には、スペルチェックが認識する単語を事前に登録するためのメカニズムが実装されています。このメカニズムにより、特定の業種、業界、団体に特有の用語に対応した特製の辞書をあらかじめ作成できるようになりました。

メインの辞書に対して、複数のカスタム辞書を同時に使用することができます。

この新しいメカニズムには、4th Dimension および 4D Write のスペルチェックが対応しています。

インストール

カスタム辞書は、Spellcheck フォルダの中にあるメイン言語のサブフォルダの中にテキスト形式のファイルを配置するだけでインストールすることができます。拡張子が「.txt」であれば、ファイル名は自由に設定することができます（例：astronomy.txt）。

メイン言語のサブフォルダは、4th Dimension 2004 に最初から存在します。その場所は次のとおりです。

■ Windows：（4D アプリケーションのフォルダ） ¥4D Extensions¥Spellcheck¥

■ Mac OS：（4D アプリケーションパッケージ） :Contents:4D Extensions:Spellcheck:

メイン言語のサブフォルダ名は、スペルチェックの主要な言語名に対応しているため、決して変更しないでください。

■ English

■ French

■ German

■ Spanish

それぞれのフォルダには、デフォルトの単語ファイルが含まれています。このファイルに単語を追加する、あるいは別の単語ファイルを追加することによって、カスタム辞書を拡張することができます。

クライアント/サーバ

スペルチェックファイルは、それぞれの 4D Client アプリケーションの中に配置されています。したがって、クライアント/サーバソリューションにカスタムスペルチェック辞書をインストールする場合、それぞれのクライアントマシンの所定の場所にカスタム辞書ファイルを配置する必要があります。

ファイルの内容

スペルチェック辞書ファイルには、次のようにキャリッジリターンコードで区切られた単語のリストを記述します。

4D

Server

Runtime

独立した単語（例：boogie）またはハイフンで連結された表現（例：boogie-woogie）は登録することができますが、スペースで区切られた一連の単語で構成された表現（例：Boogie Woogie）を登録することはできません。

Windows と Mac OS では、キャラクターセットが異なるため、各プラットフォーム用のテキストファイルを別個に用意する必要があります。ファイルはそれぞれのプラットフォームで作成し、それぞれのプラットフォームで保存するようにしてください。

辞書のロードおよび使用方法

特製の辞書ファイルは、標準の辞書と同じようにアプリケーションの起動時にロードされます。辞書の言語は、アプリケーションで使用される言語によって決まります。

セッション中、**SET DICTIONARY** コマンドを使用することにより、辞書を切り替えることができます。この場合、選択された辞書の言語フォルダにカスタム辞書が存在すれば、それも一緒にロードされます。

スペルチェックについていえば、標準の辞書に登録された単語とカスタム辞書に登録された単語による動作の違いはありません。

バックアップおよび復元

4th Dimension 2004.4 では、統合されたバックアップモジュールにいくつかの新機能が追加されました。

複数のログファイルを連続して統合する

4th Dimension 2004.4 では、複数のログファイルを連続してデータベースに統合できるようになりました。たとえば、(4回のバックアップを実行した結果) 4個のバックアップログファイルが存在する場合、最初のバックアップを復元した後、4個のバックアップログファイル続けて統合することができるようになりました。

復元されたデータベースの起動後に表示される標準の「開く」ダイアログでは、カレントログファイル (.4DL) またはログバックアップファイル (.4BL) を選択することができます。有効なログバックアップファイル (.4BL) を選択した場合、ログがデータベースに統合された後、再び「開く」ダイアログが表示されます。ここで次のログバックアップファイル (.4BL) を選択することができます。この操作は有効なカレントログファイル (.4DL) を選択するまで繰り返されます。最終的にカレントログファイルが選択されると、アプリケーションが実行を開始します。

ミラーデータベースのバックアップ

バージョン 2004.4 の 4D Server を使用すれば、論理ミラーによるバックアップシステムが稼働している場合、ミラーマシン側でデータベースのバックアップが実行できるようになりました。

注：論理ミラーによるバックアップシステムが稼働している場合、メインサーバ側でバックアップを実行することはできません。ミラーマシンとの同期が失われてしまうためです。

ミラーマシン側でバックアップを実行する場合、ファイルメニューから選択できる手動バックアップ、環境設定のバックアップスケジュールによる自動バックアップ、あるいはコマンドによるバックアップなど、どのような方法でもバックアップを開始することができます。

ログファイルの統合中、およびバックアップの実行中は、メインサーバとの同期が失われないように、ミラーマシン側のデータベースが自動的にロックされます。

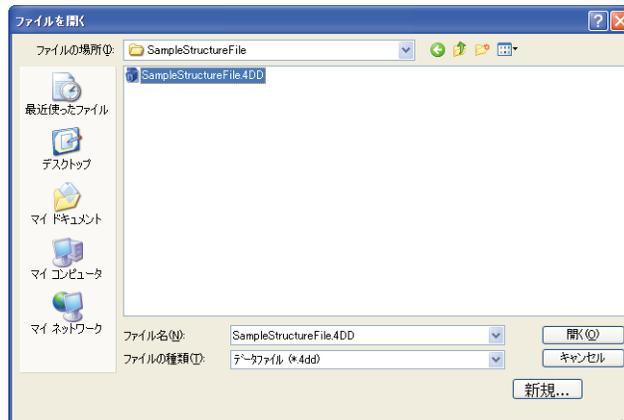
- ログファイルを統合している間は、バックアップを開始することができません。ログファイルの統合中に **BACKUP** コマンドを実行した場合、エラー 1417が発生します。
- バックアップを実行している間は、すべてのサーバプロセスが保留されるため、ログファイルの統合を開始することができません。

4D Runtime Volume License アプリケーションで復元を実行する

4th Dimension 2004.4では、コンパイル後にビルドされ、4D Runtime Volume License を組み込まれたアプリケーションで直接バックアップを復元できるようになりました。

ビルドされたアプリケーションには、「ようこそダイアログ画面」が存在しないため、バックアップを復元できるようにする場合、以前のバージョンではカスタムインタフェースで **RESTORE** コマンドが実行できるようにする必要がありました。

バージョン 2004.4では、コマンドを使用しなくても、バックアップを復元することができます。復元を実行する場合、Alt (Windows) または option (Macintosh) キーを押しながらアプリケーションを起動します。次のように標準の「データファイルを開く」が表示されます。



このダイアログ画面でバックアップファイル (.4BK) またはログバックアップファイル (.4BL) を選択すれば、バックアップを復元することができます。

注：バージョン2004.4の新しいコマンドCHECK LOG FILEを使用し、4D Runtimeアプリケーションで上記のダイアログを表示するという方法もあります（後述）。

Right-to-left 言語のサポート

4th Dimension2004.4では、Windowsで実行した場合のRight-to-left言語のサポートが強化され、カスタムモードでインタフェースオブジェクト（フォームおよびメニューバー）が自動的に反転されるアプリケーションを開発できるようになりました。

注：

- Right-to-left言語とは、アラビア語のように右から左へ記述する言語を指します。
- これに相当するMac OS Xの機能はありません。

オブジェクトを反転した場合、テキストの方向が変化するだけでなく、ピクチャ、フォーム上におけるオブジェクトの位置、ウインドウタイトル、ボタンのタイトル、メニューバーの位置なども変化します。

注：現行バージョンの4th Dimensionでは、フォームを反転して印刷することはできません。

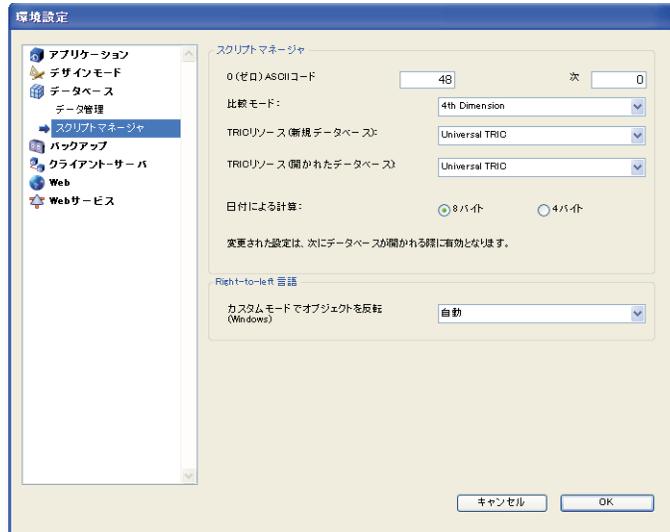
Right-to-left 言語モードを設定する

WindowsのカスタムモードにおけるRight-to-left言語モードは、ふたつのレベルで設定されます。

- データベースレベルのRight-to-left言語モードは、データベースの環境設定で設定します。
- データベースレベルのRight-to-left言語モードが有効にされている場合、フォームレベルでRight-to-left言語モードを無効にすることができます。

データベースレベルのRight-to-left 言語モード

Right-to-left言語モードは、環境設定のスクリプトマネージャページ（「データベース」テーマ）にある「カスタムモードでオブジェクトを反転（Windows）」オプションで設定します。



データベースの実行環境に応じ、次のオプションの中からRight-to-left言語モードの適用方法を選択することができます。

- **いいえ**：このオプションが選択された場合、システムの言語がRight-to-left言語であっても、データベースは一般的な言語モードで動作します。変換されたデータベースの場合、デフォルトでこのオプションが選択されています。
- **自動**：このオプションが選択された場合、システムの言語がRight-to-left言語であれば、データベースは自動的にデータベースはRight-to-left言語モードに切り替わります（例えば、アラビア語版Windowsで実行した場合）。バージョン2004.4以降の4th Dimensionで作成されたデータベースの場合、デフォルトでこのオプションが選択されています。
- **はい**：このオプションが選択された場合、システムの言語が一般的な言語モード（左から右へ）であっても、データベースはRight-to-left言語モードに切り替わります。

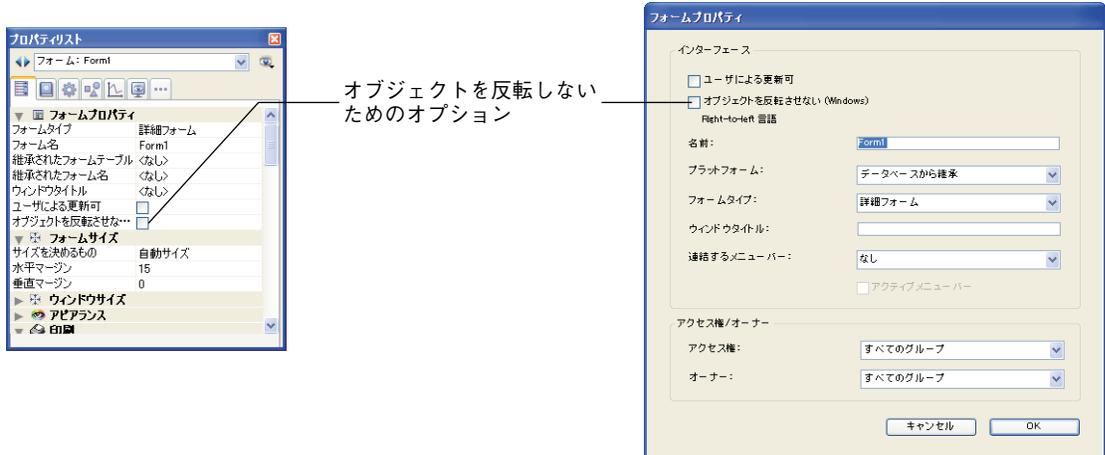
値を変更した場合、新しい設定はすぐに有効になります（環境設定ダイアログ確定後）。この設定はデータベースレベルで適用される点に留意してください。

注：Right-to-left言語モードは、コマンドでも設定することができます（後述の「SET DATABASE PARAMETER、Get database parameter」を参照してください）。

フォームレベルでRight-to-left言語モードを無効にする

Right-to-left言語モードが有効な場合、設定はフォームごとに変更することができます。このオプションを使用すれば、Right-to-left言語モードのデータベースの中に一般的な言語モード（左から右へ）のフォームを含めたりすることができます。

フォームレベルでRight-to-left言語モードを無効にするためのオプションは、プロパティリスト、および4Dエクスプローラから表示することのできる「フォームプロパティ」ダイアログで設定することができます。

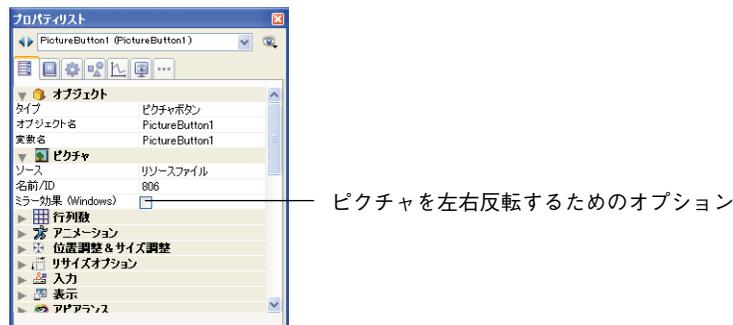


フォームのプロパティで「オブジェクトを反転しない (Windows)」が有効にされている場合、データベースの設定に関係なく、そのフォームについてはWindowsのカスタムモードでオブジェクトが反転されません。

ピクチャの反転

デフォルトの設定では、オブジェクトの位置が反転されていても、ピクチャの内容そのものは反転されません。しかし、ある種のピクチャについては、左右を反転する必要があるかもしれません。例えば、オブジェクトを指している矢印などのインターフェース画像については、オブジェクト位置の反転に応じて左右を反転する必要があります。

ピクチャの内容を反転するには、プロパティリストの「ピクチャ」テーマに追加された新しい「ミラー効果 (Windows)」オプションを使用します。



「ミラー効果」オプションは、ライブラリピクチャ、スタティックピクチャ、ピクチャフィールド、ピクチャ変数、およびピクチャを使用するフォームオブジェクト（ピクチャボタン、ポップアップメニュー、3Dボタンなど）に対して適用できます。

このオプションが有効にされている場合、画像に左右反転（ミラー効果）が適用されません。



ミラー効果が有効になるためには、次の条件が揃っていません。

- カスタムモード
- Windows
- フォームの Right-to-left モードが有効

プログラム制御

Right-to-left モードは、コマンドで制御することができ、オブジェクトの表示位置は表示モードに応じて制御することができます。

- データベースレベルの Right-to-left モードは、**SET DATABASE PARAMETER** コマンドおよび **Get database parameter** 関数を使用してアクセスすることができます。両コマンドの詳細については、後述の「SET DATABASE PARAMETER、Get database parameter」を参照してください。
- フォームレベルの Right-to-left モードは、新しい **GET FORM PARAMETER** コマンドを使用して調べることができます。このコマンドの詳細については、後述の「GET FORM PARAMETER」を参照してください。

ウインドウの座標

Right-to-left 言語に対応するためにフォームやオブジェクトを反転されている場合、ウインドウまたは画面の右上が座標の起点となります。

その場合、次のコマンドはウインドウのウインドウまたは画面の右上を起点として座標を処理します。

- Open window
- Open form window
- Open external window
- GET WINDOW RECT
- SET WINDOW RECT
- Find window

アプリケーションビルド

接続ダイアログの表示を強制する

ビルドされたクライアント / サーバアプリケーションのクライアント部分を起動する際、「4D Serverへの接続」ダイアログを強制的に表示し、任意のサーバが選択できるようになりました。ダイアログを表示するには、Alt (Windows) または option (Macintosh) を押しながらクライアントアプリケーションを起動します。

ビルドされたクライアントアプリケーションは、起動するとビルドで設定されたサーバに自動接続するため、以前のバージョンでは「4D Serverへの接続」ダイアログを表示させることができませんでした。

データファイルの場所を指定する

4th Dimension 2004.4では、アプリケーションビルドの際にデータファイルのパス名が設定できるようになりました。このオプションを使用してアプリケーションをビルドすれば、配布先でデータファイルを作成または選択しなくても、すぐに使用を開始することができます。

データファイルのパスを指定する場合、プロジェクトファイル「BuildApp.XML」の中で新しいXMLキー DataFilePath を記述しておきます。

DataFilePath

/ Preferences4D/ BuildApp/ DataFilePath

有効な値

データファイル名およびアクセスパス。

説明

ビルドアプリケーションのデータファイルパスです。パス名は、絶対パスまたはストラクチャファイル (.4DC) からみた相対パスで表記します。パス名は、次のいずれかのシンタックスで表記します。

■ マルチプラットフォーム (Posix)

■ 絶対パスの記述例：

volumes/Macintosh HD/folder/mydata.4DD

■ 相対パスの記述例：

/folder/mydata.4DD

■ Windows

■ 絶対パスの記述例：

C:\mybase\folder\mydata.4DD

■ 相対パスの記述例：

\folder\mydata.4DD

■ Mac OS (HFS)

■ 絶対パスの記述例：

Macintosh HD:folder:mydata.4DD

■ 相対パスの記述例：

:folder:mydata.4DD

アプリケーションが配布されるプラットフォームに応じ、適切なシンタックスを使用してください。

このXMLキーを省略、あるいは無効な値を渡してビルドした場合、最初の起動でWindowsの場合はストラクチャファイルと同じ階層、Macintoshの場合はアプリケーションと同じ階層にあるストラクチャ名と同名のデータファイルが選択されます。該当するデータファイルがない場合、「データファイルを選択」ダイアログが表示され、ユーザの選択したデータファイル、または新しく作成されたデータファイルが使用されます。

ランゲージ

SET DATABASE PARAMETER、Get database parameter

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

引数	タイプ	説明
table	テーブル	→ 対象テーブル、省略した場合はデフォルトテーブル
selector	倍長整数	→ セレクタ番号
value	倍長整数;文字列	→ パラメータの値

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

引数	タイプ	説明
table	テーブル	→ 対象テーブル、省略した場合はデフォルトテーブル
selector	倍長整数	→ セレクタ番号
stringValue	文字列	← 文字列パラメータの値
返り値	倍長整数	← パラメータの値

SET DATABASE PARAMETER コマンドの引数<value>には、文字列タイプが含まれるようになりました。また、**Get database parameter** 関数では、新しい文字列タイプの引数<stringValue>が利用できるようになりました。文字列タイプの引数は、セレクタ番号36で使用されます（下記を参照）。

その他、全部で4種類の新しいセレクタ定数が追加されました。

セレクタ	値	適用範囲
TCP_NODELAY	33	4Dアプリケーション
Client Server Port ID	35	4Dアプリケーション
WEDD Signature	36	データベース
Invert Objects	37	データベース

■ セレクタ=33 (TCP_NODELAY)

有効な値：0または1 (0=オプションを使用しない、1=使用する)

説明：TCP_NODELAY ネットワークオプションの使用を有効または無効に設定します。同オプションは、ネットワーク通信の最適化メカニズムを制御しているTCP/IPプロトコルの内部的な設定です。サーバマシン、クライアントマシンごとに個別の設定をすることができます。サーバマシン、クライアントマシンともにデフォルトで1 (オプションを使用する) に設定されています。

クライアント/サーバ接続をDSLまたはVPN (Virtual Private Network) 経由で確立している場合などの特定の状況においては、このオプションを無効にすることにより、顕著にアプリケーションパフォーマンスが向上するかもしれません。このオプションを変更する場合、異なるクライアント/サーバ設定で充分のテストを実施し、慎重に作業を実施する必要があります。

設定値を変更した場合、新しい設定が有効になるためには、アプリケーションを再起動する必要があります。

■ セレクタ=35 (Client Server Port ID)

有効な値：0から65535

説明：4D Serverが(4D Clientに対して) データベースを公開するために使用されるTCPポート番号を変更します。デフォルトの値は19813です。この値を変更すれば、複数の4Dクライアント/サーバアプリケーションを同じマシンで同時に使用することができます。その場合、それぞれのアプリケーションごとに異なるポート番号を設定します。

公開ポート番号は、ストラクチャファイルに記録されています。スタンドアロン版の4th Dimensionで設定することもできますが、クライアント/サーバ版で実行しなければ値に意味はありません。

設定値を変更した場合、新しい設定が有効になるためには、アプリケーションを再起動する必要があります。

■ セレクタ=36 (WEDD Signature)

有効な値：1から255バイトの文字列

説明：使用中のデータベース (ストラクチャファイルおよびデータファイル) のWEDD署名を設定します。デフォルトの値はヌルストリング (WEDDが未定義) です。WEDD署名を設定する場合、大文字と小文字が区別される点に留意してください。

WEDD 署名は、ストラクチャファイルに対して特定のデータファイルを結びつけるために使用されます。WEDD 署名を施されたストラクチャファイルは、同じ署名を施されたデータファイルと一緒になければ使用できません。WEDD 署名の詳細については、4th Dimension デザインリファレンスを参照してください。

このセレクタは、特定の署名が施されているアプリケーションのアップグレード版を配布する場合に利用することができます。

Get database parameter 関数にこのセレクタを渡す場合、新しい任意の引数 <stringValue> に WEDD 署名が返され、関数の返り値には 0 が返されます。

■ セレクタ =37 (Invert Objects)

有効な値：0、1、または 2 (0=反転しない、1=自動、2=反転する)

説明：Windows 版のカスタムモードで Right-to-left 言語に対応する場合、フォーム、フォームオブジェクトのレイアウト、メニューバーなどを反転するために使用します。詳細については、本ドキュメント 7 ページの項目「Right-to-left 言語のサポート」を参照してください。

-0 に設定した場合、システム言語に関係なく、カスタムモードでオブジェクトを反転しません (環境設定を「いいえ」にするのと同じ)。

-1 に設定した場合、システム言語が Right-to-left 言語であればカスタムモードでオブジェクトを反転し、それではなければ反転しません (環境設定を「自動」にするのと同じ)。

-2 に設定した場合、システム言語に関係なく、カスタムモードでオブジェクトを反転します (環境設定を「はい」にするのと同じ)。

テーマ：ストラクチャアクセス

GET FORM PARAMETER

GET FORM PARAMETER({table;} form; selector; value)

引数	タイプ	説明
table	テーブル	→ フォームテーブル、省略した場合はデフォルトテーブル
form	文字列	→ フォーム名
selector	倍長整数	→ パラメータコード
value	倍長整数	← フォームパラメータの値

GET FORM PARAMETER コマンドを使用すれば、<table> および <form> で指定したフォームのパラメータの値を <value> で受け取ることができます。

調べるパラメータの種類は、<selector>で指定します。渡すことができるのは、Form Parameters テーマに分類されている次の定数です。

定数	タイプ	値
NonInverted Objects	倍長整数	0

注：テーマ数を増やさずに内容を拡張するため、定数テーマ Form Options が 4th Dimension 2004.4 では Form Parameters に改名されました。

<selector>に NonInverted Objects を渡してコマンドを実行すると、<value>には Windows のカスタムモードにおけるフォームの反転表示モードが返されます。フォームの反転表示は、Right-to-left 言語に対応したアプリケーションで使用されるモードです。詳細については、前述の「Right-to-left 言語のサポート」を参照してください。

■ オブジェクトを反転させる場合、<value>には 0 が返されます。

■ オブジェクトを反転させない場合、<value>には 1 が返されます。

Windows のカスタムモード以外で実行された場合、<value>には常に 1 が返されます。

実際にフォームオブジェクトが反転されるかどうかについては、複数の要素が関係している点に留意してください。つまり、環境設定の「カスタムモードでオブジェクトを反転 (Windows)」、フォームプロパティの「オブジェクトを反転させない (Windows)」

そしてデータベースを実行しているシステムの言語が関係しています。次の表には、各要素の組み合わせと **GET FORM PARAMETER** コマンドで返される値の関係が示されています。

環境設定	フォーム プロパティ	Windowsの カスタムモード における反転	GET FORM PARAMETER コマンドで 返される値
カスタムモード でオブジェクト を反転 ¹	オブジェクトを 反転させない		
いいえ	<input type="radio"/>	<input type="radio"/>	1
		<input type="radio"/>	1
	<input type="radio"/>		1
			1
自動	<input type="radio"/>	<input type="radio"/>	1
		<input type="radio"/>	0
	<input type="radio"/>		1
			1
はい	<input type="radio"/>	<input type="radio"/>	1
		<input type="radio"/>	0
	<input type="radio"/>		1
			0

1.環境設定以外にも、SET DATABASE PARAMETER コマンド、Get database parameter関数で値の読み書きができます。

テーマ：フォーム

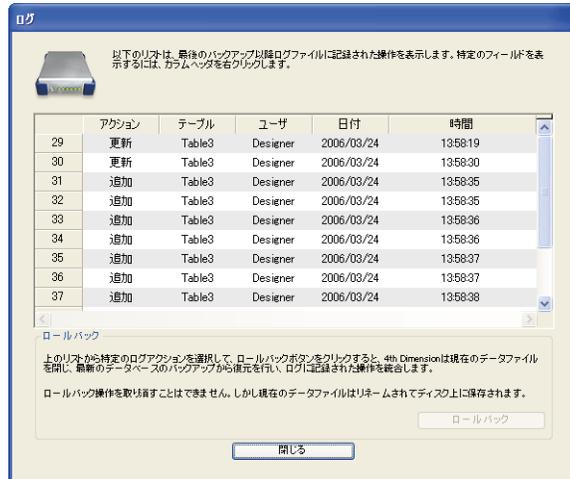
CHECK LOG FILE

CHECK LOG FILE

引数	タイプ	説明
----	-----	----

このコマンドには、引数はありません。

新しい**CHECK LOG FILE** コマンドは、4th Dimension のユーザモードでファイルメニューから「ログファイルをチェック...」を選択するのと同じ動作をするコマンドです。実行すると、データベースで使用しているログファイルの内容がブラウザできるダイアログが表示されます。



このダイアログ画面では、データファイルに対して実行された操作を取り消すことのできるロールバックボタンを使用することができます。詳しい使用方法については、4th Dimensionのユーザリファレンスを参照してください。

注：ロールバックは、比較的強力な操作であるため、CHECK LOG FILE コマンドの使用はデータベースの管理者に制限することが勧められています。

CHECK LOG FILE コマンドは、スタンドアロンアプリケーションで実行している場合に使用することができます。具体的には、4D Runtime アプリケーション（ユーザモードにアクセスできないアプリケーション）での使用を想定しています。クライアント/サーバアプリケーションで実行した場合にはエラー1421が返され、「ログ」ダイアログは表示されません。

注：サーバマシンでは、いつでもファイルメニューから「ログファイルをチェック」を選択することができます。

ログファイルを使用していないデータベースでコマンドを実行した場合にはエラー1403が返され、「ログ」ダイアログは表示されません。

トランザクション中またはインデックス処理中にコマンドを実行した場合、環境設定のバックアップ/一般設定ページ（「バックアップ」テーマ）の設定に従い、一定時間、実行開始を保留します。待機時間が経過した後、問題の操作が継続している場合、エラー1422が返され、コマンドの実行はキャンセルされます。

コマンドによって返されるエラーは、**ON ERR CALL** コマンドでエラーハンドリングメソッドをインストールすることにより、処理することができます。

テーマ：バックアップ

DOM SET XML ELEMENT VALUE

DOM SET XML ELEMENT VALUE(elementRef{;XPath}; elementValue{; *})

引数	タイプ	説明
elementRef	文字列	→ XML 要素リファレンス
xPath	テキスト	→ XML 要素の Xpath パス
elementValue	文字列;変数	→ 要素値
*	*	→ 渡した場合：CDATA 形式で値を設定

DOM SET XML ELEMENT VALUE コマンドが XPath 表記に対応するようになりました。また、CDATA で要素値を設定できるようにもなりました。

■ XPath パス：要素値を設定する際、その要素に XPath 表記でアクセスすることができません。XPath 表記でアクセスする場合、<elementRef>にはルート要素のリファレンスを渡し、任意の引数 <xPath> に要素の XPath パスを渡します。

▼ 次のような XML があるとします。

```
<Maths>
  <Postulate>1+2=3</Postulate>
</Maths>
```

<Postulate> 要素の値を変更する場合、<Maths> 要素のリファレンスがあらかじめ変数 vRoot に代入されていれば、次のように記述して要素に直接アクセスすることができます。

DOM SET XML ELEMENT VALUE(vRoot;"/Maths/Postulate";"2* 4= 8")

結果として次のように XML が書き換えられます。

```
<Maths>
  <Postulate>2* 4= 8</Postulate>
</Maths>
```

■ CDATA 形式：任意のアスタリスク (*) を渡した場合、要素値は CDATA 形式で書き込まれます。

注：CDATA 形式は、レンダリングせずにデータを書き込むために使用されます。

▼ 前述の XML の <Postulate> 要素に「12<18」というデータを書き込む場合、「<」という文字はエスケープして「<」に置換する必要があります。そうしないのであれば、データを CDATA 形式で書き込む必要があります。

<Postulate> 要素の XML ノードが vElemRef に代入されている場合、次のように記述して「12<18」というデータを書き込むことができます。

` エスケープして書き込む

DOM SET XML ELEMENT VALUE(vRefElem;"12< 18")

書き出される XML

```
<Maths>
  <Postulate>12&lt; 18</Postulate>
</Maths>
```

`CDATA形式で書き込む

DOM SET XML ELEMENT VALUE(vRefElem;"12< 18";*)

書き出される XML

```
<Maths>
  <Postulate><![CDATA[12< 18]]></Postulate>
</Maths>
```

テーマ：XML.

SET ABOUT

SET ABOUT(itemText; method)

SET ABOUT コマンドは、On Server Startup データベースメソッド、またはストアプロシージャを使用すれば、サーバマシンで実行できるようになりました。コマンドは4th Dimension や 4D Client の場合と同じように動作します。

サーバマシンで実行した場合、コマンドは新規プロセスで実行されます。

テーマ：メニュー

4D View

4D View 2004.4 では、次の操作がコマンドで実行できるようになりました。

- エリアのセル、行、列を選択解除する
- エリアをスクロールする

新しい定数 pv selection reduce

定数テーマ PV Selection action に分類される新しい定数 pv selection reduce が追加されました。この定数は、次のコマンドの引数 <action> に渡すことができます。

PV SELECT CELL

PV SELECT RANGE
 PV SELECT RANGES LIST
 PV SELECT COLUMNS
 PV SELECT ROWS

これらのコマンドに `pv selection reduce` を渡した場合、指定された領域（セル、範囲、範囲リスト、列、行）が 4D View エリア <area> における選択されたセルから外されます。

指定した領域にカレントセルが含まれていた場合、そのセルは選択が解除されるとともにカレントセルではなくなります。

▼ 次のようなメソッドを記述すれば、ユーザがセルを Alt+クリック（Windows）または option+クリック（Macintosh）するたびに、選択と選択解除を切り替えることができます。

　`範囲を選択する

PV SELECT RANGE(area;1;5;2;9;pv selection set)

　`クリックされたときにコールするメソッド

PV ON EVENT(area;pv on clicked;"ExampleView")

　`ExampleView メソッド

C_LONGINT(\$1;\$2;\$3;\$4;\$5)

If((\$2=pv on clicked) & (\$3=2048)) `Alt+ クリックまたは Option+click

If(**PV Is cell selected**(area;\$4;\$5)=1)

　　`クリックされたセルが選択されているときは解除する

PV SELECT RANGE(area;\$4;\$5;\$4;\$5;pv selection reduce)

Else `クリックされたセルが選択されていないときは選択する

PV SELECT RANGE(area;\$4;\$5;\$4;\$5;pv selection add)

End if

End if

PV SELECT ALL

PV SELECT ALL(area{; selection})

引数	タイプ	説明
area	倍長整数	→ 4D View エリア
selection	整数	→ 選択オプション：0=選択する、1=解除する

PV SELECT ALL コマンドは、すべてのセルを選択解除するためにも使用できるようになりました。

デフォルトで実行した場合、つまり引数<selection>を渡さなかった場合や0が渡された場合、4D View エリア<area>のすべてのセルが選択されます。1を渡した場合、すべてのセルが選択解除されます。

テーマ：PV Selection

新しいコマンド

PV SCROLL AREA

PV SCROLL AREA(area; horizontal; vertical; mode)

引数	タイプ	説明
area	倍長整数	→ 4D View エリア
horizontal	倍長整数	→ ピクセル値または列の番号
vertical	倍長整数	→ ピクセル値または行の番号
mode	整数	→ スクロールモード：0=絶対、1=相対、2=セル

PV SCROLL AREA コマンドを使用すれば、4D View エリア<area>の表示内容を<horizontal>および<vertical>に渡した値に応じ、プログラムでスクロールすることができます。引数には、ピクセル値で表わした移動量、あるいはセルの座標を渡します。移動の種類は、引数<mode>で指定します。

注：4D View ドキュメントに複数のペインが存在する場合、コマンドによるスクロールはカレントペインに適用されます。

■ 引数<mode>に0を渡した場合、<horizontal>および<vertical>に渡した値はピクセル値であるものとみなされ、スクロールは4D View エリアの左上セルを起点として実行されます（絶対スクロール）。

■ 引数<mode>に1を渡した場合、<horizontal>および<vertical>に渡した値はピクセル値であるものとみなされ、スクロールはその時点で4D View エリアに表示されている左上セルを起点として実行されます（相対スクロール）。

正の値を<horizontal>および<vertical>に渡した場合、右方向または下方向にスクロールすることになります。引数に負の値を渡した場合、左方向または上方向にスクロールすることになります。

注：スクロールは、左上のセルの内容が完全に表示されるように自動修正されます。

■ 引数<mode>に2を渡した場合、<horizontal>および<vertical>に渡した値はそれぞれセルの列および行の番号であるとみなされ、スクロールは指定された座標のセルが左上に表示されるように実行されます。

▼ 次のようなメソッドを記述すれば、クエリで見つかったセルが表示されるように4D View ドキュメントを自動的にスクロールすることができます。

```
PV FIND ONE(area;"Smith";1;1;$col;$row)
```

```
PV SCROLL AREA(area;$col;$row;2)
```

4D Pack

AP Add table and fields

AP Add table and fields(tableName; fieldNamesArray; fieldTypesArray; fieldLengthsArray; listFormTemplate; detailFormTemplate)) → 倍長整数

AP Add table and fields コマンドの実行中にエラーが発生した場合、発生したエラーの内容に応じて特定のエラーコードが返されるようになりました。

以前のバージョンでは、エラーの有無に関係なく、共通のエラーコード (0) が返されていました。

コマンドが返すエラーコードの種類と意味は次のとおりです。

- 1 無効なパラメータが渡された (テーブル名を省略した場合など)
- 2 同じ名前のテーブルが既に存在する
- 3 配列のタイプが不正 (<fieldNamesArray> に整数配列を渡した場合など)
- 4 配列のサイズが不正 (異なるサイズの配列を渡した場合など)
- 5 フィールドタイプが不正 (無効な値を渡した場合など)
- 6 フィールドの追加に失敗した
- 7 テーブルの作成に失敗した

