

4D v11 SQL r2

*Addendum
Windows/Mac OS*



4D v11 SQL Release 2 (11.2) - Addendum ***Windows[®] and Mac OS[®] Versions***

Copyright© 1985 - 2008 4D SAS / 4D, Inc.
All Rights Reserved..

The software and the manual are copyrighted and may not be reproduced in whole or in part except for the personal licensee's use and solely in accordance with the contractual terms. This includes copying the electronic media, archiving, or using the software in any manner other than that provided for in the Software license Agreement.

4D, 4D Draw, 4D Write, 4D View, 4th Dimension®, 4D Server and the 4th Dimension and 4D logos are registered trademarks of 4D SAS.

Windows, Windows XP, Windows Vista and Microsoft are registered trademarks of Microsoft Corporation.

Apple, Macintosh, QuickTime, and Mac OS are trademarks or registered trademarks of Apple Computer Inc.

Mac2Win Software is a product of Altura Software, Inc.

ICU Copyright © 1995-2008 International Business Machines Corporation and others. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

4D includes cryptographic software written by Eric Young (ey@cryptsoft.com)

4D includes software written by Tim Hudson (tjh@cryptsoft.com).

Spellchecker © Copyright SYNAPSE Développement, Toulouse, France, 1994-2008.

ACROBAT © Copyright 1987-2008, Secret Commercial Adobe Systems Inc. All rights reserved. ACROBAT is a registered trademark of Adobe Systems Inc.

All other referenced trade names are trademarks, registered trademarks, or copyrights of their respective holders.

目次

Chapter 1	概要	7
Chapter 2	デザインモード	9
	新しい Welcome ウィザード	10
	新しいフィールドプロパティ	11
	自動インクリメント	11
	内部格納オプション	12
	新しいキーワード定義アルゴリズム	13
	新しいサーバで実行メソッド属性	14
	実行コンテキスト	15
	ポインタ	15
	例	15
	リストボックス	16
	非表示行	16
	複数行の入力と表示 (Windows)	17
	リソースエクスプローラ	17
	概要	18
	利用	18
	クライアント / サーバモードでの動作	24
	Web エリア	27
	Web エリアの作成	27
	Mac OS 上での注意点	28
	Windows 上での注意点	29
	割り当てられる変数	29
	標準アクション	31
	フォームイベント	31
	ユーザインターフェース	33
	ランゲージコマンド	33
	クライアント / サーバアプリケーションのビルド	33
	データベース接続の保存	35
	ファイルの作成	36
	利用	37
	Mac OS で Web サーバを開始する	37
Chapter 3	ランゲージ	39
	On SQL Authentication データベースメソッド	39
	ツール	41
	OPEN WEB URL	41

4D 環境	42
OPEN ADMINISTRATION WINDOW	42
NOTIFY RESOURCES FOLDER MODIFICATION	44
Get 4D folder	44
SET DATABASE PARAMETER, Get database parameter	46
システム環境	49
LOG EVENT	49
リストボックス	50
GET LISTBOX ARRAYS	50
セレクション	50
TRUNCATE TABLE	50
ピクチャ	51
BLOB TO PICTURE	51
READ PICTURE FILE	52
クエリ	53
QUERY SELECTION WITH ARRAY	53
QUERY BY FORMULA, QUERY SELECTION BY FORMULA	53
Web サービス (クライアント)	55
AUTHENTICATE WEB SERVICE	55
SET WEB SERVICE OPTION	55
Web サービス (サーバ)	56
SOAP DECLARATION	56
レコードロード時トリガ	58
Web エリア	58
WA OPEN URL	58
WA Get current URL	59
WA OPEN BACK URL	60
WA Back URL available	60
WA OPEN FORWARD URL	61
WA Forward URL available	61
WA REFRESH CURRENT URL	61
WA STOP LOADING URL	62
WA Execute JavaScript	62
WA EXECUTE JAVASCRIPT FUNCTION	63
WA SET URL FILTERS	63
WA GET URL FILTERS	66
WA SET EXTERNAL LINKS FILTERS	66
WA GET EXTERNAL LINKS FILTERS	68
WA GET LAST URL ERROR	69
WA Get last filtered URL	70
WA GET URL HISTORY	70
WA Create URL history menu	71
WA Get page title	72
WA SET PAGE CONTENT	73
WA Get page content	74

	WA SET PAGE TEXT LARGER	74
	WA SET PAGE TEXT SMALLER	74
	WA SET PREFERENCE	75
	WA GET PREFERENCE	76
Chapter 4	4D Server v11 SQL	77
	アーキテクチャ	77
	リモートフォルダの管理	78
	新しいサーバ管理ウィンドウ	80
	モニタ	81
	ユーザ	84
	プロセス	87
	メンテナンス	91
	アプリケーションサーバ	94
	SQL サーバ	96
	HTTP サーバ	97
	新しいリモート機能	100
	クライアントマシンによる 4D Server の管理	100
	リモートマシンからのコンパイル	101
	リモートマシンからバックアップ環境設定にアクセスする	103
	4D Server v11 SQL とランゲージ	103
	サーバ上でフォーミュラによるコマンドの実行	103
	命名セレクションとセット	105
	サーバ設定	105
	TCP ポート番号	105
	Maintenance & Security Center へのアクセス	106
Chapter 5	プラグイン	107
	4D Internet Commands	107
	SMTP_SetPrefs	107
	4D View	108
	ペインの固定と固定解除	108
	新しいコマンド	109
	PV FREEZE PANES	109
	PV UNFREEZE PANES	110
	ランゲージの変更	111

1

概要

4D v11 SQL release 2 によろこそ。

この新しいバージョンでは 4D Server v11 SQL が提供され、4D クライアント / サーバアプリケーションで、4D v11 SQL エンジンのパワーとパフォーマンスの恩恵を得ることができます。さらに 4D Server v11 SQL は新しいユーザフレンドリな管理ウィンドウの実装と大きな最適化が施されました。

また 4D v11 SQL r2 にはシングルユーザおよびクライアント / サーバに適用される、多くの新しい機能が追加されました：新しいフィールドおよびメソッドプロパティ、新しいリソースエクスプローラ、フォームに置くことのできる Web エリアなど。統合された 4D ランゲージにも新しい機能や変更が加えられました。

最後に、4D Internet Commands と 4D View プラグインのバージョン 11.2 にも新しい機能が加えられました。

これらすべての新しい特徴は以下の章で説明されています：

- デザインモード
- ランゲージ
- 4D Server v11 SQL
- プラグイン

2

デザインモード

この章では、データベースレベル、RAD ツール、およびフォームにおいて、4D v11 SQL のデザイン環境に対して行われた新しい機能の追加や変更について説明します。

- 新しい Welcome ウィザード
- 新しいフィールドプロパティ
- 新しい "サーバ上で実行" メソッド属性
- リストボックスの非表示行と複数行への入力
- リソースエクスプローラ
- 新しい "Web エリア" オブジェクト
- クライアント / サーバアプリケーションビルドに関連する新しい機能
- データベースへの接続を保存する新しいモード
- Mac OS 上で Web サーバを開始する新しいモード

新しい Welcome ウィザード

4D に新しい Welcome ウィザードが追加されました。このウィザードはデフォルトで 4D アプリケーションが起動される際に表示されます：



このウィザードは、プログラムを開始する際、新しい 4D ユーザをガイドする目的で加えられました。ウィザードを使用して、アプリケーションの使用やアクティベーションなど、利用可能なすべてのアクションにアクセスできます。アクションの隣にある **INFO** ボタンをクリックすると、より詳細な情報が表示されます。

デフォルトで、ウィザードは起動のたびに表示されます。起動時にこのダイアログを表示しないオプションをチェックすると、このダイアログを表示しないようにできます。ヘルプメニューの Welcome ウィザードコマンドを選択して、いつでもこのダイアログを再び表示し、またこのオプションを非選択にできます：



新しいフィールドプロパティ

4D v11 SQL のバージョン 11.2 ではデータベースフィールドに新しい 2 つのプロパティが追加されました。

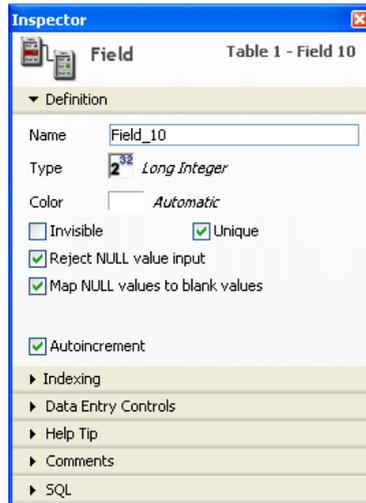
自動インクリメント 4D に新しいフィールドプロパティ "自動インクリメント" が追加されました。これはレコードのユニークな ID を生成するために使用できます。

このプロパティが設定されたフィールドは、テーブルにレコードが作成されるたびに自動でインクリメントされます。これは 4D エンジンによってのみ管理されます。このフィールドにユーザやプログラムが入力したり更新したりすることはできません。更新を行おうとすると、エラーが生成されます。

割り当てられた数値は永続的であり、レコードが削除されたとしても再利用されることはありません。トランザクション中に生成された番号も、トランザクションがキャンセルされると失われ、再利用されません。

このプロパティは整数、倍長整数、および Integer 64 bits タイプのフィールドで使用できます。これは SQL 属性の "AUTO_INCREMENT" に対応するものであり、SQL 言語を使用して指定することができます (例を参照)。

4D の場合、"自動インクリメント" 属性はストラクチャエディタのインスペクタに表示され、変更することができます：



この属性が設定されたフィールドには、インスペクタの "SQL" 情報エリアにも AUTOINCREMENT ラベルが表示されます。

同一のテーブル内に複数の "自動インクリメント" フィールドを定義できます。それぞれのフィールドごと個々にカウンタが割り当てられます。フィールドからプロパティが取り除かれると、カウンタは 0 にリセットされます。

- ▶ 以下のコードは、"animals" テーブルに倍長整数タイプの "id" フィールドを作成します：

```

ARRAY TEXT ($names;6)
$names{1}:="dog"
$names{2}:="cat"
$names{3}:="penguin"
$names{4}:="whale"
$names{5}:="butterfly"
$names{6}:="ostrich"
Begin SQL
CREATE TABLE animals(
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(30) NOT NULL,
  PRIMARY KEY (id));
INSERT INTO animals (name) VALUES (:$names);
End SQL

```

以下のコードを実行すると：

```
SELECT * FROM animals;
```

結果は以下のようになります：

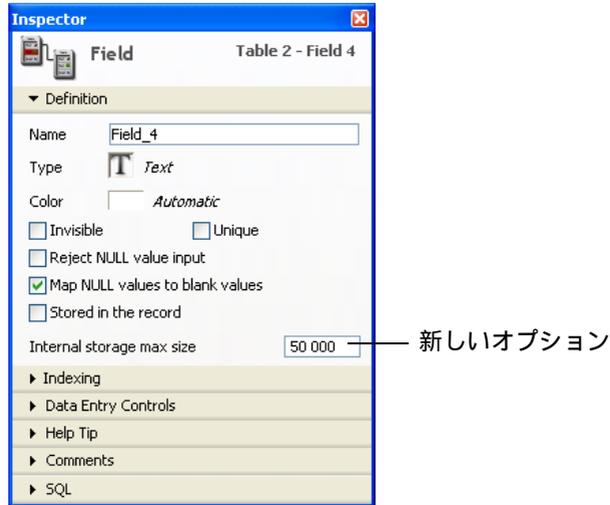
id	name
1	dog
2	cat
3	penguin
4	whale
5	butterfly
6	ostrich

内部格納オプション

4D v11 SQL の以前のバージョンでは最適化のため、BLOB、ピクチャ、およびテキスト（オプション）タイプのフィールドのデータをレコードの外部に格納していました。

バージョン 11.2 では、データをレコードの外側に格納するかどうかを、サイズに基づき選択できるようになりました。限定されたサイズの BLOB、ピクチャ、およびテキストデータをアプリケーションで扱っている場合、レコードにこれらを含めたほうが良いかもしれません。

これはフィールドインスペクタの新しいオプションで設定できます：



" 内部格納最大サイズ " エリアにカスタム値を入力します。値はバイト単位で入力し、サイズがこの値以下であればデータはレコード内部に格納されます。例えば、ピクチャフィールドに 30 000 を指定すると、20 KB のピクチャはレコード内に格納され、40 KB のピクチャはレコードの外部に格納されます。

デフォルト値は 0 で、すべてのデータはレコード外部に格納されます。

新しいキーワード定義アルゴリズム

バージョン 11.2 より、キーワードインデックスを作成する際、4D はテキストフィールドの単語を判断するために ICU ライブラリに基づく新しいアルゴリズムを使用します。

以下のアルゴリズムと、以下の点において異なる点に留意してください：

- 小数点を含め、数値は全体でひとつの語とみなされます。
- アポストロフィやハイフンを含む単語は、それらを含めひとつの語とみなされます。

注 詳細は以下のページを参照してください：<http://www.icu-project.org/userguide/boundaryAnalysis.html>.

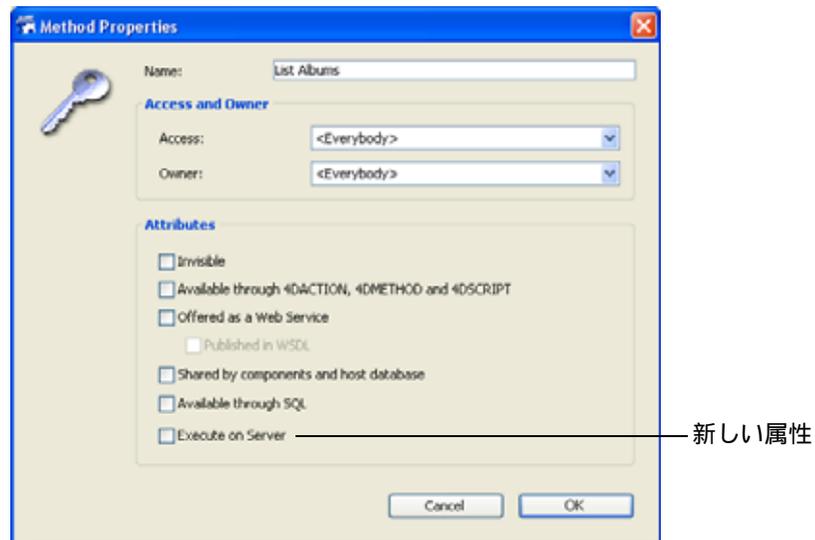
以下の例は、以前のバージョンと 4D v11.2 での、キーワード定義の差異を示します。インデックスするテキストは "Today's price is \$11.42" です。:

以前のアルゴリズム (4D v11.0, 11.1)	新しいアルゴリズム (4D v11.2 以降)
Today	Today's
s	price
price	is
is	11.42
11	
42	

警告: 既存のデータベースにこの方式を適用するには、新しいバージョンの 4D でインデックスを再作成しなければなりません。

新しいサーバで実行メソッド属性

4D v11 SQL Release 2 にはプロジェクトメソッドに新しい属性、サーバ上で実行が追加されました。この属性はメソッドプロパティダイアログボックスのほか、メソッド属性の一括設定を使用して設定できます:



この属性はクライアント / サーバモードの 4D アプリケーションでのみ考慮されます。

このオプションにチェックされていると、どのように呼ばれたかにかかわらず、プロジェクトメソッドは常にサーバ上で実行されます。

実行コンテキスト

プロジェクトメソッドの実行コンテキストはトリガのそれと同じです。サーバ上のメソッドは、レコードのロックのため、またはクライアント側のコンテキストに対応するトランザクションのため、同じデータベースコンテキストを共有します。

メソッドのすべての引数 (\$1, \$2, ...) はサーバに送られ、戻り値 \$0 が使用されていけばクライアントに返されます。

Execute on server コマンドと異なり、このオプションはサーバ上にプロセスを作成しません。4D Server は実行をリクエストしたクライアントプロセスに対応するサーバ上のプロセスを使用します。

さらに、このオプションはサーバ上でのメソッド実行を簡略化します。通常の方法であるかのように、引数は双方向で自動的に転送されるからです。Execute on server コマンドの場合は、メソッドはサーバ上で非同期に実行されるため、結果を読みだすためにはセマフォなどを利用して追加のプログラミングをしなければなりません。

ポインタ

変数 (変数、配列、または配列要素) へのポインタを渡すと、ポイントされた値がサーバに送信されます。ポイントされた値がサーバ上のメソッドで変更されると、変更された値がクライアントに返され、クライアント側の対応する変数の値が書き換えられます。

テーブルやフィールドへのポインタは参照として送信されます (テーブル番号、フィールド番号)。カレントレコードの値は自動では交換されません。

注 このオプションはインタプリタモードとコンパイルモードで同様に動作します。

例

これは "サーバ上で実行" 属性を持つ Myappli プロジェクトメソッドのコードです :

```
C_POINTER($1) ` テーブルへのポインタ
C_POINTER($2) ` フィールドへのポインタ
C_POINTER($3) ` 配列へのポインタ
C_TEXT($4) ` 検索する値
C_LONGINT($0) ` 結果
```

```

` 検索を行いレコードごとの値を戻す
QUERY($1-> ; $2-> = $4)
While(Not(End selection($1->)))
  APPEND TO ARRAY($3-> ; myFormula($1))
  NEXT RECORD($1->)
End while
UNLOAD RECORD($1->)
$0 := Records in selection( $1->)

```

クライアント側では、メソッドを以下のように呼び出します：

```

ARRAY TEXT(myArray; 0)
$Vlnum := MyAppli (->[Table_1]; ->[Table_1]Field_1; ->myArray; "to find")

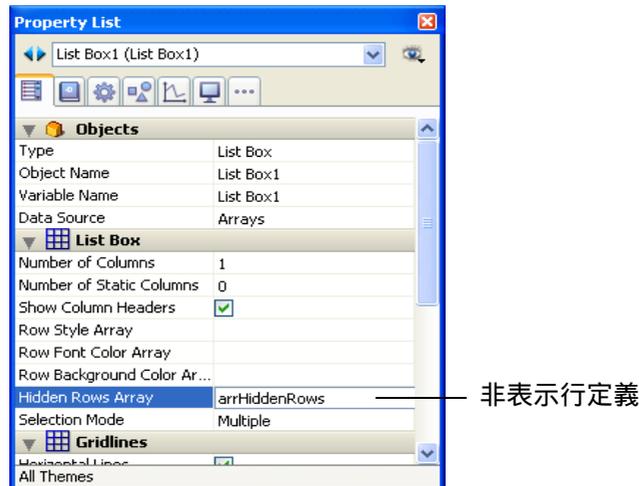
```

リストボックス

非表示行

配列タイプのリストボックスで、一つ以上の行を非表示にできるようになりました。

行を非表示に設定する方法は、スタイルやカラーを設定する方法と同様です。"リストボックス"の新しい非表示行配列プロパティを使用して、リストボックスに配列を割り当てます。



ブールタイプの配列を指定します。この配列はリストボックスと同じ要素数を持っていなければなりません。配列のそれぞれの要素は、リストボックスの対応する行の表示 / 非表示を表します：

- 行を隠すには、True を渡す。

- 行を表示するには、False を渡す。

配列が指定されていないか、配列が空の場合、リストボックスのすべての行が表示されます。デフォルトで、追加された行は表示されます。

行の非表示はリストボックスの表示にのみ影響します。隠された行は依然として配列に存在し、プログラムで管理することができます。ランゲージコマンド、特に Get number of listbox rows や GET LISTBOX CELL POSITION は、表示 / 非表示の状態を考慮しません。例えば 10 行あるリストボックスで 9 行隠されていると、Get number of listbox rows は 10 を返します。

ユーザから見ると、リストボックス中の非表示行を視覚的に認識することはできません。例えばすべてを選択コマンドを使用したとしても、表示行だけを選択することができます。

GET LISTBOX ARRAYS コマンドは、非表示行配列を返すように変更されました ([ページ 50](#)、"[GET LISTBOX ARRAYS](#)" コマンドを参照)。

複数行の入力と表示 (Windows)

Windows 版の 4D v11 SQL r2 では、同じリストボックスの "セル" に、複数行のテキストを入力および表示できます。

改行を追加するには Ctrl+Carriage return を押します：

Last Name	First Name	City
Electra	Omar	Nevers
Elizabeth	Oliver	Paris, TX (USA)

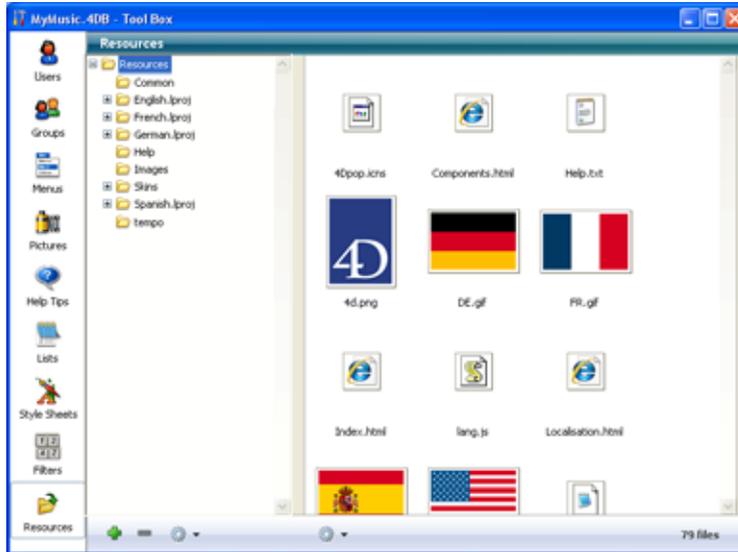
行の高さは自動では変更されないことに注意してください。

リソースエクスプローラ

4D v11 SQL では、リソースアーキテクチャが、データベースストラクチャファイル (.4db または .4dc) と同階層にある Resources フォルダに基づくよう変更されました。このフォルダはデータベースやコンポーネントのすべての "リソース" を格納するために使用されます。言い換えれば翻訳、アプリケーションインターフェースのカスタマイズ (ピクチャ、テキスト、XLIFF など)、あるいは動作に必要なすべてのファイルが置かれます。

クライアント / サーバアーキテクチャでこのフォルダの管理を容易にするため、4D v11 SQL バージョン 11.2 では、Resources フォルダの内容を管理するために使用することのできる新しいツール、リソースエクスプ

ローラが組み込まれました。このツールは 4D のツールボックスにあります：



概要

リソースエクスプローラは、カレントデータベースの Resources フォルダの内容を階層リストで表示します。このツールにはフォルダの内容の管理を容易にするいくつかの機能、追加、削除、プレビュー等があります。

リソースエクスプローラは 4D のローカルモードまたはリモートモードで使用できます。

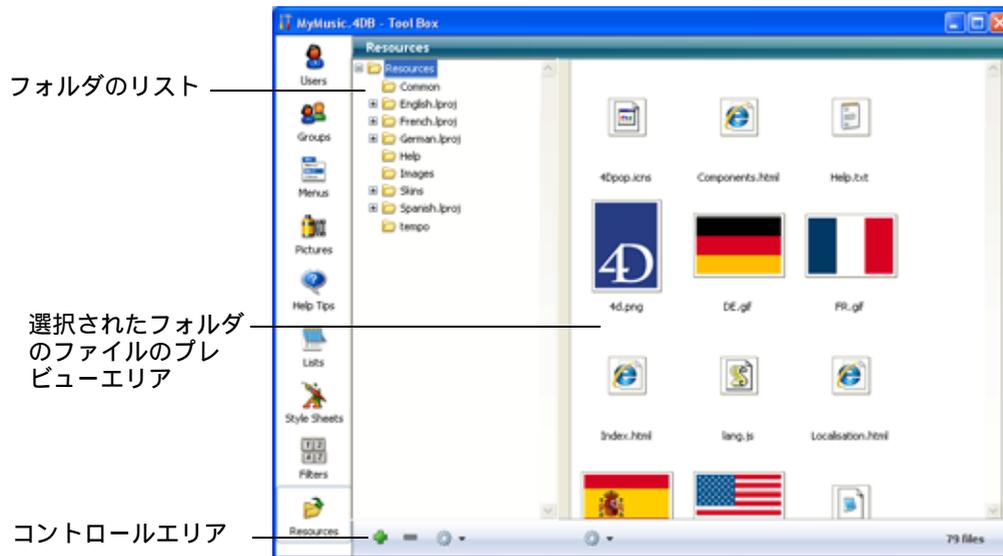
リモートモードのとき、リソースエクスプローラは最も便利です。このコンテキストでは、4D Server に接続されたすべてのリモートマシン間で、Resources フォルダの内容を同期させるために使用できます。言い換えれば、リソースエクスプローラは、クライアント / サーバ環境でリソースの共有を管理するために使用できます。

通知メカニズムにより、Resources フォルダの内容が変更されると、クライアントマシンにそれが知らされます。そしてそれぞれのクライアントマシンはサーバとの同期を行います。

利用

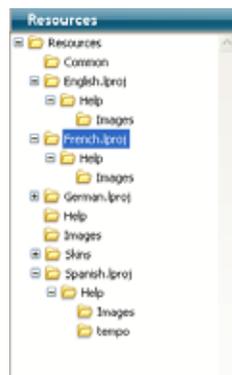
リソースエクスプローラを、カレントデータベースの Resources フォルダの内容を表示したり、変更したりするために使用できます。リソース要素を追加、削除、検索、そして表示するための様々なコントロールを持っています。

リソースエクスプローラには二つのエリア、フォルダのリストとプレビューエリアがあります。それぞれ独自のコントロールボタンがあります：



フォルダのリスト

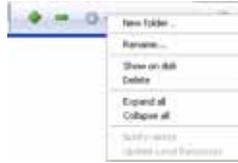
フォルダのリストエリアには、データベースの Resources フォルダの中にあるフォルダの階層が表示されます。



フォルダを展開したり折りたたんだり、選択したりできます。フォルダ名をクリックすると、含まれるすべてのファイルが、リソースエクスプローラ右側のプレビューエリアに表示されます。

ボタンやアクションメニューを含むコントロールエリアは、ウィンドウの下側にあります。

フォルダのリストの下にあるコントロールエリアには、[+] と [-] ボタン、およびアクションメニューがあります。



[+] は、選択されたフォルダ内にフォルダを作成します。フォルダが選択されていない場合はトップレベルに作成します。[-] ボタンは選択されたフォルダとその内容物を削除します。

注 フォルダリストのコンテキストメニューを使用することもできます。このメニューのコマンドは、アクションメニューと同じものです。

アクションメニューのコマンドは以下のとおりです：

- **新規フォルダ ...**: 選択されたフォルダの中に、またはフォルダを選択していない場合トップレベルに、フォルダを作成します。このコマンドを選択するとダイアログが表示され、作成するフォルダの名称を指定できます。フォルダはディスク上に物理的に作成されるため、(: や / のような) システムが許可しない文字を使用しないようにしなければなりません。
- **名称変更 ...**: 選択されたフォルダの名称を変更するためのダイアログが表示されます。

注 Resources フォルダの名称を変更することはできません。

- **ディスク上に表示 (ローカルモードでのみ有効)**: オペレーティングシステムのウィンドウにフォルダを表示します。
- **削除**: 選択したフォルダとその内容を削除します。

注 Resources フォルダを削除することはできません。

- **すべて展開 / すべて折りたたむ**: リストのすべてのフォルダを展開したり折りたたんだりします。
- **クライアントに通知 (リモートモードでのみ有効)**: Resources フォルダの内容の更新を他のクライアントマシンに通知するために使用します。他のクライアントはこの通知を受け取ると、他のクライアントは個々の設定に基づき即座にまたは後に、Resources フォルダを同期します。([ページ 25 "クライアントマシンのアップデート"](#) の節を参照)

このコマンドは、Resources フォルダに変更を行った後、他のクライアントに即座に同期を行わせたい場合に使用します。

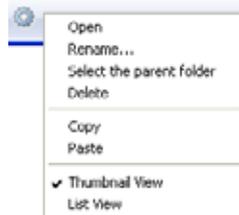
- ローカルリソースを更新 (リモートモードでのみ有効): サーバマシンの Resources フォルダと、ローカルのそれとの同期を強制したい場合に使用します。

このコマンドはサーバマシンの Resources フォルダが更新されたことを知らせる通知を受け取った際に使用します (ページ 25、" クライアントマシンの通知 " の節参照)。またローカルリソースを更新を選択する際に Shift キーを押したままにすると、ローカルの Resources フォルダの内容をグローバルに更新できます (サーバフォルダをダウンロードし、ローカルフォルダと置き換える)。

プレビューエリア

リソースエクスプローラの右側のエリアはプレビューエリアです。ここには選択されたフォルダおよびサブフォルダの内容が表示されます。このエリアはフォルダリストで選択された項目が変わるたびに更新されます。

このエリアの下にはアクションメニューがあり、特定の操作が可能です :



- 開く : 選択したファイルをデフォルトのエディタで開きます (存在すれば)。このアクションはファイルをダブルクリックしたのと同じです。
- 名称変更 : 選択したファイルを名称変更するためのダイアログが表示されます。
- 親フォルダを選択 : フォルダリスト中で、選択したファイルの親フォルダを選択します。このコマンドは、Resources フォルダ階層のなかで、ファイルの位置を知るために使用できます。実際、プレビューエリアには、サブフォルダに含まれるものを含む、すべてのファイルが表示されます。
- 削除 : 選択されたファイルを削除します。
- コピー : 選択されたファイルをクリップボードにコピーします。
- ペースト : クリップボードにピクチャまたはファイルパスが格納されているとき、クリップボードの内容を選択したフォルダにペーストします。クリップボードにピクチャが格納されている場合、4D は対応するタイプ

のピクチャを作成します。作成されるファイルの名称を指定するためのダイアログが表示されます。

- サムネイル表示 / リスト表示 : これらのコマンドは切り替えて表示されま
す。
- サムネイル表示モードでは、選択されたフォルダのファイルはサムネ
イルで表示されます。ピクチャタイプのファイルはプレビューされ
(4Dがピクチャフォーマットを認識できる場合)、他のファイルタイプ
ではシステムアイコンが使用されます。

サムネイルとして表示



このプレビューモードでは、コンテキストメニューからファイル編集
コマンドにアクセスできます (後述)。

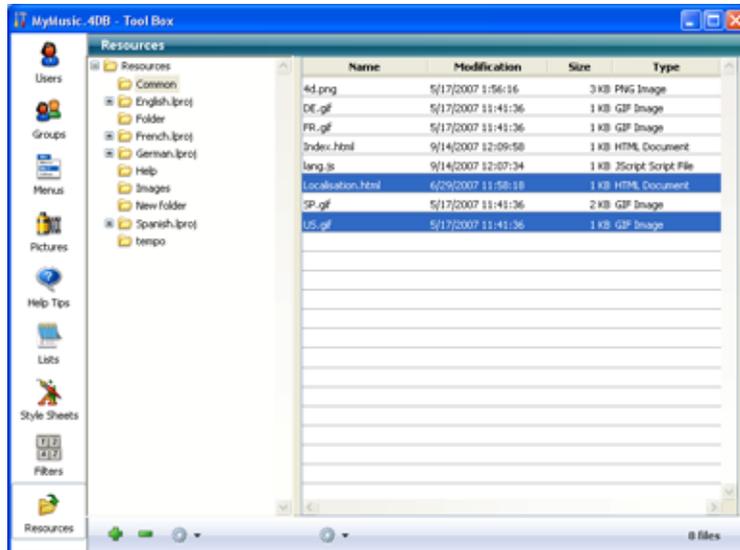
それぞれのピクチャに表示されるヘルプ Tip には、相対パス名、名称、
タイプ、更新日と時刻、サイズなどの追加情報が表示されます。



- リスト表示モードでは、選択されたフォルダのファイルは表として表
示されます。それぞれのファイルごとに、名称、更新日と時刻、サイ

ズ、およびタイプが直接表に表示されます。

リスト表示



列のヘッダエリアをクリックして表を並び替えることができます。

- コンテキストメニュー
ファイルがサムネイルで表示されているとき、エリアにコンテキストメニューを表示させることができます：



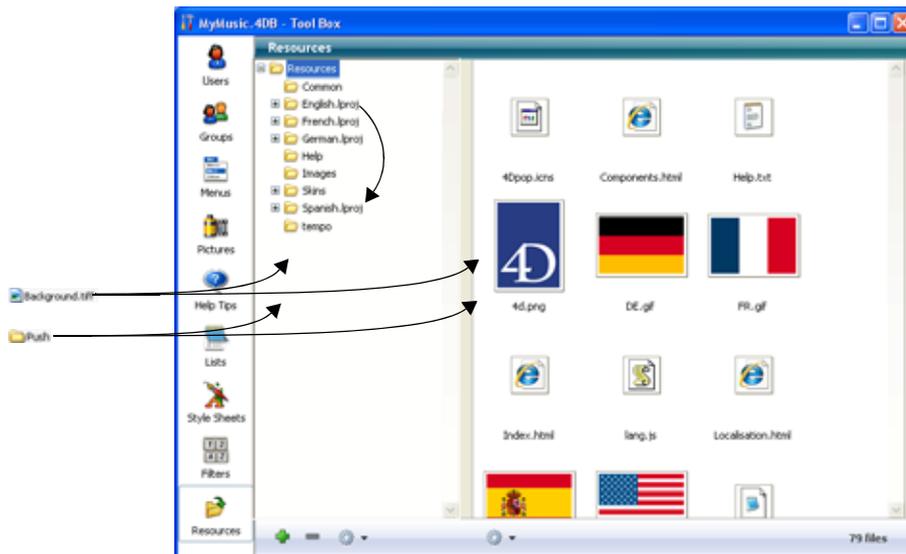
このメニューのコマンドはアクションメニューのものと同じです。ただしコンテキストに関連するコマンドのみが表示されます。

ドラッグ&ドロップの管理

リソースエクスプローラを使用して、Resources フォルダやサブフォルダにドラッグ&ドロップで項目を追加できます：

- フォルダリスト中でドラッグ&ドロップにより、Resources フォルダの階層構造を変更できます。
- またオペレーティングシステムのデスクトップからドラッグ&ドロップで、Resources フォルダにファイルを追加できます。4D は自動でドロップされたファイルのコピーを Resources フォルダに作成します。フォルダリストまたはプレビューエリアにファイルをドロップできます。

- Resources フォルダにフォルダをドロップできます。フォルダはオペレーティングシステムまたは他の 4D アプリケーションの Resources フォルダにあります。フォルダが他の 4D アプリケーションからドロップされる場合、それは移動されます。そうでなければ、ファイルの場合と同じようにコピーが作成されます。
フォルダをフォルダリストまたはプレビューエリアにドロップできます。後者のケースでは、プレビューエリアに表示されている階層レベルに対応する場所にフォルダが作成されます。
- Resources フォルダのピクチャをフォームエディタやメソッドエディタにドラッグ&ドロップできます。ピクチャは参照として挿入されます。



クライアント / サーバモードでの動作

リソースエクスプローラはクライアント / サーバ環境での共同開発を容易にします。特に接続されたマシンの Resources フォルダの内容を、リアルタイムで同期させるために使用できます。

注 以前のバージョンの 4D と同様、接続されていないクライアントの場合、Resources フォルダの同期は接続時に自動で行われます。

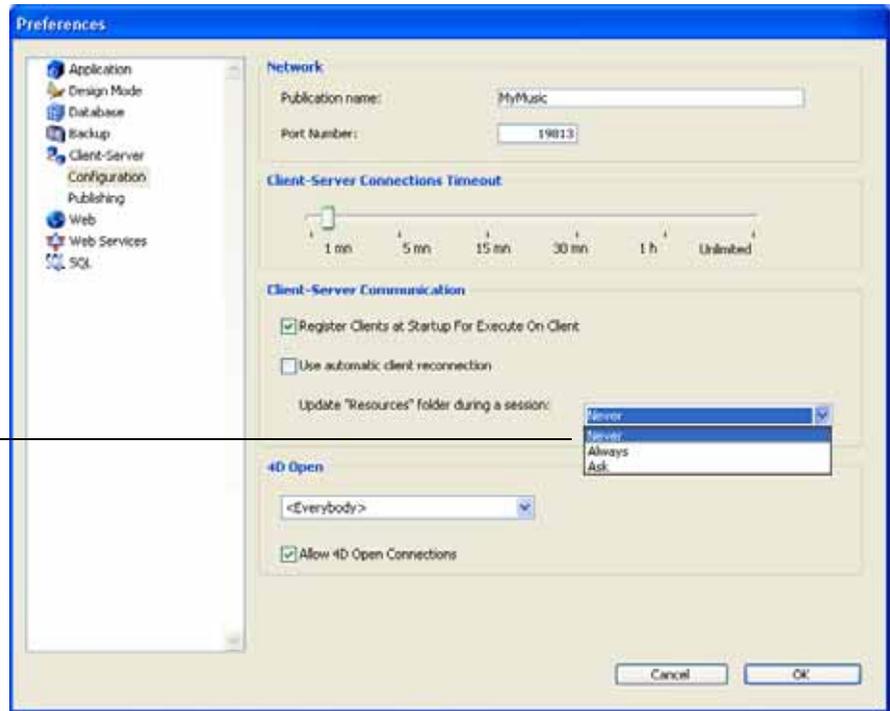
同期の主なステップは以下のとおりです：

- 1- クライアントマシンで更新を行った場合、サーバの Resources フォルダをアップデートする。
- 2 - 接続したすべてのユーザに通知する。
- 3 - 接続したクライアントマシンでリアルタイムに更新が行われる。

サーバの更新	<p>このモードは以下のように動作します: リソースエクスプローラで Resources フォルダにローカルで実行された更新は、自動でサーバに送られます。</p>
クライアントマシンの通知	<p>接続したクライアントマシンは、Resources フォルダが変更された旨の通知を受け取ります:</p> <ul style="list-style-type: none">■ クライアントにより行われた最後の更新から 2 分後にサーバから自動で (この遅延は、多数のファイルをコピーしている場合などの、不要な通知を防ぎます)。■ または更新を行った元のクライアントマシン上で、リソースエクスプローラのアクションメニューのクライアントに通知コマンドを使用して (ページ 19、"フォルダのリスト" の節を参照)。■ または新しいコマンド <code>NOTIFY RESOURCES FOLDER MODIFICATION</code> を使用する。このコマンドはストアドプロシージャを使用してサーバマシン上で Resources フォルダを更新した場合に便利です。 <p>クライアント側では、更新に関する通知を受け取った際、それぞれのマシンの設定に基づいて処理されます (次の節を参照)。</p>
クライアントマシンのアップデート	<p>サーバ上で Resources フォルダが更新された通知を受け取ったら、それぞれのクライアントマシンは同期を行うことができます。この同期は、データベースのグローバル設定、またはクライアントごとの設定に基づき、自動又は手動で行われます。</p> <p>この設定は環境設定でグローバルに、または <code>SET DATABASE PARAMETER</code> コマンドを使用してクライアントごと及びセッションごと個々に設定できます。</p>

アップデートモードの設定は、環境設定のクライアント - サーバ / 設定ページの "セッション中に "Resources" フォルダをアップデート" オプションを使用して行います :

Resources フォルダの動的アップデート設定



3 つの選択肢があります :

- **しない** : セッション中に Resources フォルダはアップデートされません。サーバからの通知は無視されます。ローカルの Resources フォルダはローカルリソースを更新コマンドを使用して手動でアップデートできます (ページ 19、" フォルダのリスト" の節を参照)。
- **常に行う** : サーバから通知が送信されると、セッション中にローカル Resources フォルダが自動で同期されます。
- **確認する** : サーバから通知が送信されると、クライアントマシン上でダイアログが表示されます。ユーザはローカル Resources フォルダの同期を受け入れるか拒否するかを選択できます。

注 設定がサーバ上の環境設定で行われると、すべてのクライアントマシンに設定が適用されます。クライアントマシン上で設定されると、設定はそのマシンでのみ有効です。

SET DATABASE PARAMETER による同期モード設定についての詳細は、[ページ 46](#)、"[SET DATABASE PARAMETER, Get database parameter](#)" コマンドを参照してください。

Web エリア

4D のバージョン 11.2 では、フォームに新しいタイプのオブジェクト、Web エリアを追加できるようになりました。このエリアには 4D 環境の中で、すべてのタイプの Web コンテンツを表示させることができます：スタティックおよびダイナミックコンテンツを含む HTML ページ、ファイル、ピクチャ、Javascript、Flash、PDF¹、そして MS Office ドキュメント (MS Office がインストールされた Windows のみ)。

注 この機能は 4D Web 2.0 Pack に含まれる 4D Live Window を置き換えます。

いくつかの新しい標準アクションと数多くのランゲージコマンドを使用して、開発者はこの Web エリアをコントロールできます。特定の変数を使用して、エリアと 4D 環境間で情報を交換できます。つまり、フォームに基本的な Web ブラウザを組み込むことができるのです。

実行中、Web エリアはコンテキストメニューを持つことができ、またドラッグ&ドロップ機能もあります。

Web エリアの描画エンジンは、アプリケーションが実行されるプラットフォームにより異なります：

- Mac OS X では、4D は Apple WebKit エンジンを使用します (Safari ブラウザの描画と同じもの)。
- Windows では、4D は ActiveX Web Controls を使用します (Internet Explorer ブラウザの描画と同じもの)。

Web エリアの作成

Web エリアは、4D フォームエディタのオブジェクトバーにある、プラグインエリア / サブフォームボタンに追加された新しい項目を使用して作成します：



1. PDF ドキュメントを表示するために、Mac OS は (システムに含まれる) プレビューを使用します。Windows では Adobe Reader が必要です。

フォームに Web エリアを追加するには、Web エリアボタンを選択して、エリアを描画します。同じフォーム上に複数の Web エリアを作成できません。

他のダイナミックフォームオブジェクトのように、Web エリアにはプログラムでの処理に使用するためのオブジェクト名、変数名を持ちます。Web エリアオブジェクトに割り当てられる標準変数はテキストタイプです。すなわち SET VISIBLE や MOVE OBJECT コマンドを Web エリアに対して使用できます。

注 Web エリアに関連付けられたテキスト変数には参照は含まれていません。そのためメソッドの引数として渡すことはできません。例えば、MyArea という名前の Web エリアで、以下のコードは使用できません：

```
MyMethod(MyArea)
```

MyMethod コード：

```
WA REFRESH CURRENT URL($1)` 動作しない
```

このようなプログラムを行うには、ポインタを使用する必要があります：

```
MyMethod(->MyArea)
```

MyMethod コード：

```
WA REFRESH CURRENT URL($1->)` 動作する
```

Mac OS 上での注意点 Mac OS 上で Web エリアを使用する際は、システムに関連する特別な条件があります。リクエスト URL にはプロトコルを必ず含めなければなりません。またエリアを含むウィンドウは "compositing" モードでなければなりません。

URL に含まれるプロトコル Mac OS 上の Web エリアで、プログラムにより処理される URL はプロトコルで開始されていなければなりません。つまり "www.mysite.com" ではなく "http://www.mysite.com" 文字列を渡さなければならないということです。

Compositing モード Web エリアを表示させるためには、エリアを "compositing モード" のウィンドウに置かなければなりません。Mac OS のこの内部的なウィンドウ処理モードは、すべての 4D ウィンドウで使用されているわけではありません。

4D v11 SQL では、"compositing モード" のウィンドウは：

- Open form window コマンドで生成されるすべてのウィンドウ
- Compositing Mode (定数値 4096) タイプを持つ、Open window コマンドで生成されたウィンドウ

- デザインモードで ( ボタンで実行された)、プロジェクトフォームを表示するウィンドウです。

注 特定の前世代のオブジェクトは compositing モードと互換がありません (例えば 4D Chart エリア)。このようなオブジェクトが compositing モードのウィンドウに表示されると、オブジェクトは動作しません。

Windows 上での注意 点

Web エリアと Web サーバのコンフリクト Windows では、Web エリアから、同じ 4D アプリケーションで起動されている Web サーバへのアクセスはお勧めできません。これを行うとコンフリクトが発生し、アプリケーションがフリーズすることがあります。もちろん他の 4D から 4D Server の Web エリアにアクセスすることはできます。自身の Web サーバにアクセスできないということです。

MSO ce ドキュメント Windows では、Web エリアを使用して、Microsoft Office ドキュメントを表示したり、編集したりできます (Microsoft Office がマシンにインストールされている場合)。特に Word、Excel、Powerpoint ドキュメントなど (.doc、.xls、.ppt などの拡張子) を処理できます。MS Office XML フォーマットもサポートされています。

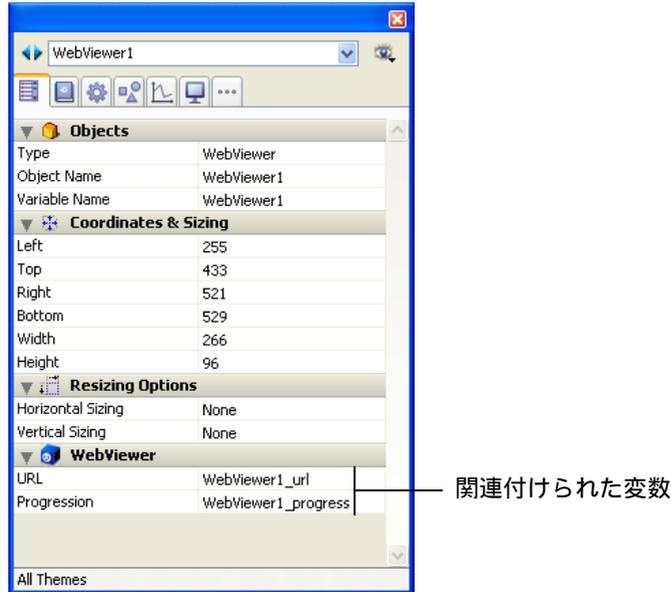
注 MS Office 2007 はデフォルトで Web ブラウザでのドキュメントの表示を許可していません。ドキュメントは常に新規ウィンドウに表示されます。以下のアドレスで説明されている方法で、この動作を変更できます：
<http://support.microsoft.com/kb/162059/ja-jp>

最後に、Windows では、Web エリアを使用して、ローカルや外部のフォルダを表示できます (ftp:// プロトコルやネットワークパス名 /myserver /myvolume を使用して)。

割り当てられる変数 標準のオブジェクト変数に加え (ページ 27 "Web エリアの作成" の節参照)、2 つの変数が自動でそれぞれの Web エリアに関連付けられます：

- "URL" 変数
- "Progression" 変数

デフォルトで、これらの変数は `areaName_url` と `areaName_progress` になります。必要であれば名称を変更できます。これらの変数にはプロパティリストからアクセスできます：



URL 変数

"URL" は文字タイプの変数です。この変数には Web エリアにロードされた URL またはロード中の URL が格納されます。

変数と Web エリア間の連携は双方向で行われます：

- ユーザが新しい URL を変数に割り当てると、この URL は自動で Web エリアにロードされます。
- Web エリアでブラウズが行われると、自動で変数の内容が更新されます。

このエリアはブラウザのアドレスバーのように機能します。Web エリアの上側にテキストエリアを置いて、内容を表示させることができます。

- URL 変数と WA OPEN URL コマンド
URL 変数は WA OPEN URL コマンドと同じ効果をもたらします ([ページ 58](#)、"[WA OPEN URL](#)" コマンドを参照)。しかしながら以下の違いに注意してください：
 - ドキュメントにアクセスする場合、この変数は RFC 準拠 ("file:///c:/My%20Doc") な URL のみを受け付け、システムパス名 ("c:/MyDoc") は受け付けません。WA OPEN URL コマンドは両方の記法を受け付けます。

- URL 変数が空の文字列の場合、Web エリアは URL をロードしません。WA OPEN URL コマンドはこの場合エラーを生成します。
- URL 変数にプロトコル (http, mailto, file など) を含まない場合、Web エリアは "http://" を付加します。WA OPEN URL コマンドは付加しません。
- Web エリアがフォーム上で表示されていない場合 (フォームの他のページに Web エリアがある場合等)、WA OPEN URL コマンドを実行しても効果はありません。一方、URL 変数に値を代入すると、カレントの URL が更新されます。

Progression 変数

"Progression" は倍長整数タイプの変数です。この変数には 0 から 100 までの値が格納され、この数値は Web エリアに表示されるページのロードされたパーセンテージを表します。

この変数は 4D が自動で更新します。手動で変更することはできません。

標準アクション

自動で Web エリアを管理するためにいくつかの新しい標準アクションが提供されます。これらのアクションはボタンやメニューコマンドに関連付けることができ、基本的な Web インターフェースを素早く実装することを可能にします。

- 前の URL を開く : このアクションは、ユーザにより行われた一連のブラウジング中、前の URL を開きます。ユーザが一つのページしかエリアに表示していない場合、関連付けられたボタンやメニューコマンドは選択不可になります。
- 次の URL を開く : このアクションは、ユーザにより行われた一連のブラウジング中、次の URL を開きます。次の URL がない、つまりユーザが一連のブラウジングで前に戻ったことがなければ、関連付けられたボタンやメニューコマンドは選択不可になります。
- 現在の URL を再読み込み : このアクションは Web エリアの現在の内容を再読み込みします。
- URL のロードを停止 : このアクションは、Web エリアのカレント URL のページやオブジェクトのロードを停止します。

フォームイベント

Web エリアの様々な状況をデベロッパがコントロールできるようにするために、4D に新しいフォームイベントが追加されました。また Web エリアは既存のいくつかのフォームイベントと互換があります。

新しいイベント

Web エリアで新しいフォームイベントを使用できます。

- On Begin URL Loading
このイベントは、Web エリアに新しい URL のロードを開始した際に生成されます。Web エリアに関連付けられた "URL" 変数を使用して、ロード中の URL を知ることができます。

注 ロード中の URL はカレントの URL と異なります (WA Get current URL コマンドの説明を参照)。

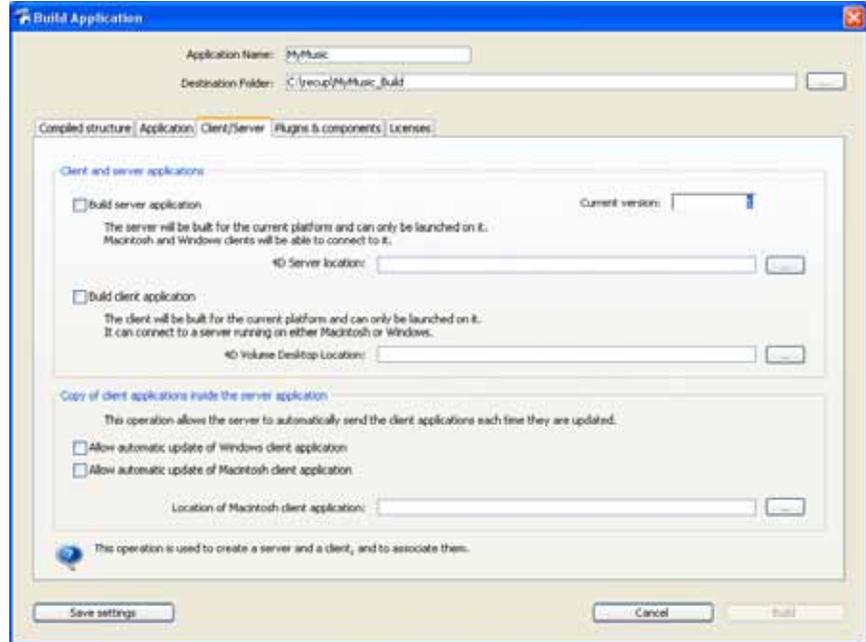
- On URL Resource Loading
このイベントは、現在の Web ページに新しいリソース (ピクチャやフレームなど) がロードされるたびに生成されます。
エリアに関連付けられた "Progression" 変数により、ロードの現在の状況を知ることができます。
- On End URL Loading
このイベントは、現在の URL のすべてのリソースがロードされた後に生成されます。
[WA Get current URL](#) コマンドを呼び出して、ロードされた URL を知ることができます。
- On URL Loading Error
このイベントは、URL のロード中にエラーを検知すると生成されます。
[WA GET LAST URL ERROR](#) コマンドをコールして、エラーに関する情報を取得できます。
- On URL Filtering
このイベントは、WA SET URL FILTERS コマンドで設定されたフィルタにより、Web エリアが URL のロードをブロックした場合に生成されます。
[WA Get last filtered URL](#) を使用してブロックされた URL を知ることができます。
- On Open External Link
このイベントは、[WA SET EXTERNAL LINKS FILTERS](#) の設定のため Web エリアによって URL のロードがブロックされ、その URL がカレントのシステムブラウザによって開かれた際に生成されます。
[WA Get last filtered URL](#) を使用してブロックされた URL を知ることができます。
- On Window Opening Denied
このイベントは、ポップアップウィンドウが Web エリアによってブロックされた際に生成されます。
[WA Get last filtered URL](#) を使用してブロックされた URL を知ることができます。

- 互換のあるイベント 以下のフォームイベントは Web エリアで使用できます：
- On Load
 - On Unload
 - On Getting Focus
 - On Losing Focus
 - On Drag Over
 - On Drop
 - On Begin Drag Over.
- ユーザインターフェイス フォームが実行されると、他のフォームエリアと対話することを可能にする、標準のブラウザインターフェイス機能が Web エリア中で利用可能になります：
- 編集メニューコマンド：Web エリアにフォーカスがあるとき、選択された内容に基づき、編集メニューコマンドを使用してコピーやペースト、すべてを選択などのアクションを実行できます。
 - コンテキストメニュー：Web エリアで、システムの標準コンテキストメニューを使用できます。コンテキストメニューの表示は [WA SET PREFERENCE](#) コマンドを使用してコントロールできます。
 - ドラッグ&ドロップ：4D のオブジェクトプロパティに基づき、ユーザは Web エリア上で、または Web エリアと 4D フォームオブジェクト間で、テキストやピクチャ、ドキュメントをドラッグ&ドロップできます。
- ランゲージコマンド プログラムで Web エリアをコントロールするためのコマンドが 20 以上追加されました。これらのコマンドは "Web エリア" テーマで見つけることができます。これらのコマンドは [ページ 58](#)、"[Web エリア](#)" の節で説明しています。

クライアント / サーバアプリケーションのビルド

以前のバージョンと同様、4D v11 SQL でも、マルチプラットフォームで自動更新オプションを持つ、カスタムクライアント / サーバアプリケーションをビルドすることができます。この操作はローカルモードの 4D で行います。

4D アプリケーションアーキテクチャの進化を反映させるため、アプリケーションビルドのインターフェースが変更されました：

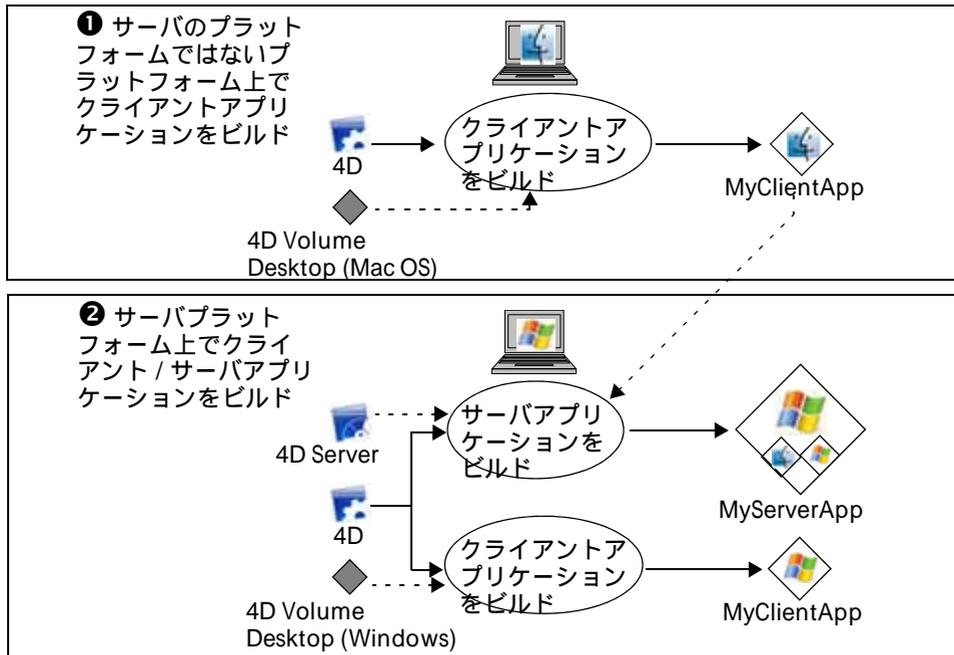


- サーバアプリケーションをビルド：ビルドフェーズでアプリケーションのサーバ部をビルドするには、このオプションにチェックします。ディスク上の 4D Server アプリケーションの位置を指定します。バージョン番号に基づきリモート接続をコントロールするために、現在のバージョンを指定できます。
- クライアントアプリケーションをビルド：ビルドフェーズでアプリケーションのクライアント部をビルドするには、このオプションにチェックします。以前のバージョンの 4D と異なり、4D Volume Desktop アプリケーションの場所を指定します。このバージョンからは、4D Volume Desktop にクライアントアプリケーションを作成するために必要な要素が含まれています。
- Windows/Macintosh クライアントアプリケーションの自動更新を許可する：ネットワーク経由でクライアント / サーバアプリケーションの自動更新を有効にするには、これらのオプションにチェックします。

マルチプラットフォームのクライアントアプリケーションを作成するには、他のプラットフォームに対応するクライアントアプリケーションのフォルダ (Mac OS ではパッケージ) をディスク上で指定します。例えば、

Windows 上でアプリケーションをビルドする場合、[...] ボタンを使用して、Mac OS 上で生成されたクライアントアプリケーションパッケージを選択します。もちろんクライアントアプリケーションは事前に対応するプラットフォーム上で生成しなければなりません。

自動更新付のマルチプラットフォームクライアント - サーバアプリケーションをビルド (Windows サーバ)



データベース接続の保存

4D v11 SQL では、4D データベースを開いたり接続したりすることを自動化するためのパラメタを含む、データベース接続ファイルを作成できます。通常接続ファイルにはリモートデータベースのアドレスや接続 ID が保存され、ユーザのいくつかの操作を省略することができます。

これらの接続ファイルは、4D Server で公開されているデータベースへの接続または、ローカルデータベースを開くために使用できます。

注 以前のバージョンの 4D では、この機能は 4D Server データベースへの接続に限定されていて、4D Client の接続ダイアログのボタンから利用できました。ファイルには .pth 拡張子が付けられていました。これらのファイルは 4D v11 SQL ではサポートされません。

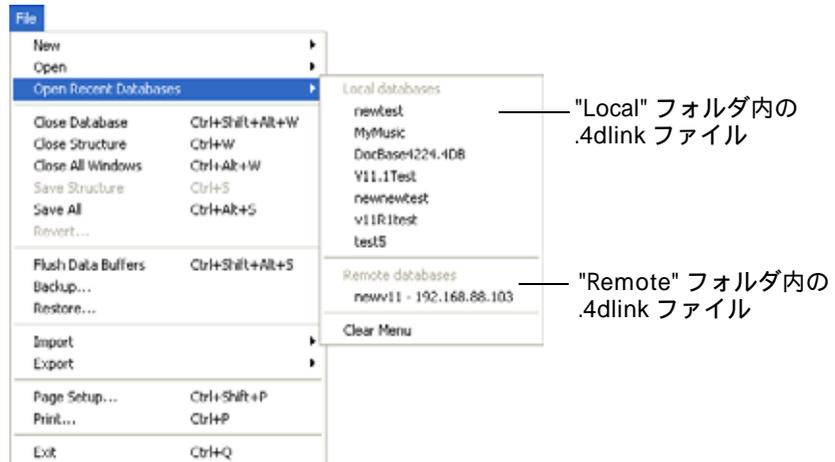
ファイルの作成

4D v11 SQL の接続ファイルは XML ファイルで、".4DLink" 拡張子が付けられます。4D は "最近使用したデータベース" のサブメニューを構築するために、このタイプのファイルを生成して利用します。ローカルデータベースを初めて開くとき、またはサーバに接続するとき、4D は .4DLink ファイルを自動で生成します。

4D が自動で作成した .4DLink ファイルは、ユーザのローカルな環境設定フォルダに置かれます。このフォルダには、2 つのディレクトリ Local と Remote が作成されます。Local フォルダにはローカルデータベースへの接続に使用できる ".4DLink" ファイルが、Remote フォルダにはリモートデータベースへの接続に使用できる ".4DLink" ファイルが置かれます：

- Windows Vista: C:/Users/UserName/AppData/Roaming/4D/Favorites v11 /
- Windows XP: C:/Documents and Settings/UserName/Application Data/4D/Favorites v11 /
- Mac OS: Users/UserName/Library/Preferences/4D/Favorites v11/

このディレクトリに置かれたファイルは、4D のファイルメニューの最近使用したデータベースを開く > サブメニューに表示されます：



".4DLink" ファイルは XML エディタを使用して作成し、接続 ID やデータベースを開くモードなどをカスタマイズした情報を含めることもできます。

".4DLink" ファイルを構築するために使用できる XML キーを定義した DTD が 4D より提供されます。この DTD は database_link.dtd という名前で、4D の /Resources/DTD/サブフォルダに見つけることができます。

利用

.4DLink 接続ファイルは 4D アプリケーションを起動し、ターゲットの 4D データベースを開くために使用できます。利用には 3 つの方法があります：

- ダブルクリックまたは 4D アプリケーションへのドラッグ&ドロップ。
- 4D の開くダイアログで選択する。
- 最近使用したデータベースを開くサブメニュー（ローカルの環境設定に置かれたファイル）。

リモートデータベースタイプの .4DLink は、他のマシンにコピーして使用することもできます。

Mac OS で Web サーバを開始する

Mac OS X では、Web アプリケーション用に予約されたポート (0 から 1023 ポート) を使用するには特別なアクセス権が必要です。マシンの "root" ユーザだけがこれらのポートを使用してアプリケーションを起動できます。

以前のバージョンの 4D では、Web サーバを起動する際に一時的な "root" セッションを開くことで、アプリケーションの起動を可能にしていました。しかし、最新の Mac OS X (Leopard) におけるセキュリティポリシーの変更のため、このメカニズムは動作しなくなりました。

4D のバージョン 11.2 からは、HelperTool という名前の、ポートを開くための特別なアプリケーションを使用します。このアプリケーションは適切なアクセス権を持っています。

このアプリケーションは 4D ソフトウェアに含まれています。これはシステムの特定の場所にインストールされなければなりません。マシン上で 1024 未満のポートを最初に開く際に、自動でインストールされます。ユーザにはツールがインストールされる旨通知され、マシンの管理者名とパスワードの入力が求められます。この操作は一回だけ必要です。

アプリケーションは "com.4D.HelperTool" と名称変更され、フォルダ "/Library/PrivilegedHelperTools/" にインストールされます。インストール後、4D Web サーバの開始や終了が透過的に可能になります。

注 このメカニズムは Mac OS X 10.4.6 以降で利用できます。これより前のバージョンの OS では、他の方法を使用する必要があります。

3

ランゲージ

この章では、4D v11 DQL のプログラムランゲージに関する新しい機能や変更についてまとめています。

On SQL Authentication データベースメソッド

新しい On SQL Authentication データベースメソッドを使用して、統合された 4D の SQL サーバに送信されたリクエストをフィルタできます。このフィルタはログイン名やパスワード、また (追加で) ユーザの IP アドレスに基づき行えます。開発者は独自のユーザテーブルや、4D のそれを使用して、接続を識別できます。接続が認証されたら、CHANGE CURRENT USER コマンドを使用して、4D データベース内のリクエストのアクセスをコントロールできます。

On SQL Authentication データベースメソッドは、存在すれば、SQL サーバへ外部からの接続が行われた時、4D または 4D Server によって自動で呼び出されます。従って 4D ユーザを管理する内部システムは有効になっていません。データベースメソッドが \$0 に True を返すと接続は受け入れられ、そうでなければ拒否されます。

注 ODBC LOGIN(SQL INTERNAL;\$user;\$password) は内部的な接続であるため、On SQL Authentication データベースメソッドを呼び出しません。

このデータベースメソッドは最大 3 つのテキストタイプの引数 (\$1, \$2, \$3) を 4D から受け取り、ブールタイプの値を返します (\$0)。これらの引数の説明は以下のとおりです：

引数	型	説明
\$1	テキスト	← ユーザ名
\$2	テキスト	← パスワード
\$3	テキスト	← (オプション) リクエスト送信元のクライアント IP アドレス


```

`$2: パスワード
`{$3: クライアントの IP アドレス }
ON ERR CALL ("SQL_error")
If (checkInternalIP($3))
  `checkInternalIP メソッドは IP アドレスが内部からかチェックする
  If ($1="victor") & ($2="hugo")
    CHANGE CURRENT USER("sql_user","")
    If (OK=1)
      $0:=True
    Else
      $0:=False
    End if
  Else
    $0:=False
  End if
Else
  $0:=False
End if

```

ツール

OPEN WEB URL

OPEN WEB URL (url{; *})

このコマンドの用途は以下のように拡張されました：

- Unicode の完全サポート
- url 引数に URL の他、ファイルパス名を渡せるようになりました。
- Mac OS で、追加のプロトコルがサポートされました (file:、mailto:、news:、http:、その他)。

アプリケーションはまず url 引数をファイルパス名として解釈を試みません。コマンドは Mac OS ではコロン (":") を、Windows では円マーク ("/") を、または file:// から始まる POSIX URL を受け入れます。この場合、4D はファイルを適切なアプリケーションを使用して開くよう、システムにリクエストします (例えば .html ファイルではブラウザ、.doc ファイルの場合は MS Word、等)。この場合 * 引数は無視されます。

以上のケースでない場合、標準のメカニズムを使用して URL が開かれず。

- ▶ 異なるタイプの URL を開くために以下の例を使用できます：

```
OPEN WEB URL("http://www.4d.com")
```

```
OPEN WEB URL( "file://C:/Users/Lauren/Documents/pending.htm")
OPEN WEB URL( "C:/Users/Lauren/Documents/pending.htm")
OPEN WEB URL( "mailto:john_martin@4d.com")
```

- ▶ この例はアプリケーションを起動するために使用できません :

```
$file:=Select document("","");0
If (OK = 1)
  OPEN WEB URL(Document)
End if
```

注 より機能を明確に表現するため、OPEN WEB URL は "Web サーバ" テーマから " ツール " テーマに移動されました。

4D 環境

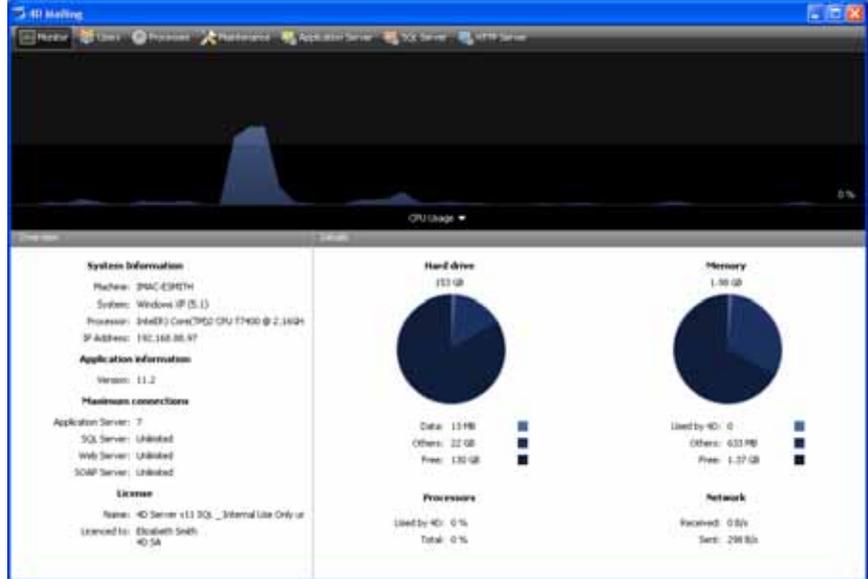
OPEN ADMINISTRATION WINDOW

OPEN ADMINISTRATION WINDOW

引数	型	説明
		このコマンドは引数を必要としません

新しい **OPEN ADMINISTRATION WINDOW** コマンドは、コマンドを実行したりリモートの 4D クライアントマシン上でサーバ管理ウィンドウを開きます。4D Server の管理ウィンドウで現在のパラメタを表示させたり、さまざまなメンテナンス操作を行ったりできます ([ページ 80](#)、" **新しいサーバ管理ウィンドウ** " の段落参照)。

4D Server のバージョン 11 から、このウィンドウをクライアントマシン上で表示できるようになりました。



このコマンドは、4D Server に接続した 4D アプリケーションから呼び出さなければなりません。以下の場合、コマンドは何も行いません：

- ローカルモードの 4D アプリケーションから呼び出されたか、サーバのストアドプロシージャとして実行された場合。
- Designer でも Administrator でもないユーザが実行した場合（この場合エラー -9991 が生成されます）。

コマンドが正しく実行されると、OK システム変数に 1 が、そうでなければ 0 が設定されます。

- ▶ この例を管理ボタンに割り当てることができます：

```

If (Application type=4D local mode)
  OPEN SECURITY CENTER
  ...
End if
If (Application type=4D remote mode)
  OPEN ADMINISTRATION WINDOW
  ...
End if
If (Application type=4D Server)
  ...
End if

```

NOTIFY RESOURCES FOLDER MODIFICATION

NOTIFY RESOURCES FOLDER MODIFICATION

引数 型 説明
このコマンドは引数を必要としません

新しい NOTIFY RESOURCES FOLDER MODIFICATION コマンドを使用して、接続されたすべての 4D マシンに、Resources フォルダが更新された旨の通知の送信を、4D Server に強制することができます。Resources フォルダの管理に関する詳細は、[ページ 16](#)、"[Resources Explorer](#)" の段落を参照してください。

更新が行われたという情報のみが送信されます。それぞれのリモートマシンは、ローカル環境設定に従い処理を行います：同期しない、自動で同期する、または警告を表示する ([ページ 24](#)、"[Notification of Client Machines](#)" の段落参照)。

このコマンドは特に、サーバ上のストアドプロシージャで Resources フォルダを更新した後、リモートマシンで同期を管理するために使用できます。

Get 4D folder

Get 4D folder ({{folder}}; {*)} → 文字列

Get 4D folder コマンドは 2 つの新しいセレクトと、Active 4D Folder や Licenses Folder の場所を決定する新しい原則を受け入れます。

新しいセレクト

Get 4D folder コマンドは folder 引数に 2 つの新しいセレクトを受け入れます。以下の定数が "4D Environment" テーマに追加されました：

定数	型	値
Logs Folder	倍長整数	7
HTML Root Folder	倍長整数	8

- Logs Folder: この定数を渡すと、Get 4D folder はカレントデータベースのログファイルが置かれるフォルダのパス名を返します。Logs という名前のこのフォルダはストラクチャファイルと同階層にあります。

Logs フォルダには以下のログファイルが置かれます：

- データベース変換
- Web サーバリクエスト
- データの検証と復元
- ストラクチャの検証と修復
- バックアップ / 復元操作のジャーナル,
- コマンドデバッグ

- 4D Server リクエスト (クライアントマシンおよびサーバで生成)
- HTML Root Folder: この定数を渡すと、Get 4D folder はデータベースのカレントの HTML ルートフォルダパス名を返します。HTML ルートフォルダは、4D Web サーバが、リクエストされた Web ファイルやページを検索するフォルダです。デフォルトで、このフォルダはストラクチャファイルと同階層 (ローカルモードの 4D や 4D Server の場合)、または 4D アプリケーションと同階層 (リモートモードの 4D) に置かれた WebFolder という名前のフォルダです。この場所は環境設定の Web/ 設定ページまたは SET HTML ROOT コマンドで動的に変更できます。Get 4D folder がリモートの 4D から呼び出されると、返されるパスはリモートマシンのものであり、4D Server のものではありません。

Active 4D Folder の場所 Get 4D folder コマンドは Active 4D Folder セレクタにより、ユーザ環境設定フォルダを返します。

以前のバージョンでは、このフォルダはライセンスを含むことができ、4D アプリケーションのタイプや一連の 4D のインストールにより、場所が異なりました。

今バージョンから、ユーザ環境設定フォルダは 4D v11 の環境設定、Macros v2、そして Favorites v11 フォルダと shortcuts.xml を含み、常に以下の場所に置かれます：

- (Windows Vista) C: \Users \{user_name} \AppData \Roaming \4D
- (Windows XP) C: \Documents and settings \{user_name} \Application Data \4D
- (Mac OS X) /users/{user_name}/library/Preferences/4D

Licenses Folder の場所 Licenses Folder の場所は以下となります：

- (Windows Vista) C: \ProgramData \4D \Licenses
- (Windows XP) c: \Documents and settings \all users \Application Data \4D
- (Mac OS X) /library/application support/4D/licenses

4D Volume Desktop がマージされたアプリケーションの場合、ライセンスフォルダはアプリケーションのパッケージに含まれます。

4D の以前のバージョンと同様、権限の不足からシステムにライセンスフォルダを作成できない場合、フォルダは以下の場所に作成されます：

- (Windows Vista) C: \Users \{user_name} \AppData \Roaming \4D
- (Windows XP) c: \Documents and settings \{user_name} \Application Data \4D

- (Mac OS X) /users/{user_name}/library/Application Support/4D

注 (サーバデータベースからダウンロードした要素が置かれる) リモート接続のキャッシュフォルダは常に以下の場所に作成されます:

- (Windows Vista) C:/Users/{user_name}/AppData/Local/4D
- (Windows XP) C:/Documents and Settings/{user_name}/Local Settings/Application Data
- (Mac OS X) /users/{user_name}/library/Caches/4D

このフォルダは、ユーザプロファイルの移行の一部として移行はされません。

SET DATABASE PARAMETER, Get database parameter

いくつかの新しいセレクタ (定数) が SET DATABASE PARAMETER と Get database parameter コマンドに追加されました。

Selector = 45 (Client Log Recording)

- 値: 0 または 1 から X (0 = 記録しない、1 から X= ファイル名に付加するシーケンシャル番号)。
- 説明: コマンドを実行した 4D クライアントマシンにより送信された標準リクエストの記録を開始 / 停止する (Web リクエストを除く)。デフォルトで、値 0 は (リクエストを記録しない) です。4D はクライアントマシンによるリクエストをログファイルに記録します。このメカニズムが有効にされると、クライアントマシン上のデータベースのローカルフォルダの Logs サブフォルダに、2 つのファイルが作成されます。ファイルの名称は "4DRequestsLog_X" と "4DRequestLog_ProcessInfo_x" で、X はログファイルのシーケンシャル番号です。4DRequestLog ファイルサイズが 10MB に達すると、ファイルは自動で閉じられ、新しいファイルがインクリメントされた番号を付けられ作成されます。同じ名前のファイルが存在する場合、そのファイルは置き換えられます。value 引数を使用して開始番号を設定できます。

このテキストファイルには、タブ区切りフォーマットで、時間、プロセス番号、リクエストサイズ、処理時間などそれぞれのリクエストに関する様々な情報が格納されます。この情報は特に、アプリケーションの開発フェーズや統計を取る際に有効です。

Selector = 46 (Query By Formula On Server)

- 値: 0 (データベース設定を使用する)、1 (サーバで実行) または 2 (クライアントで実行)
- 説明: パラメタで渡したテーブルに対する QUERY BY FORMULA と QUERY SELECTION BY FORMULA コマンドの実行場所。データベースをクライアント / サーバモードで実行しているとき、

フォーミュラを使用するクエリコマンドはサーバ上あるいはクライアントマシン上で実行できます：

- 4D v11 SQL で作成されたデータベースでは、これらのコマンドはサーバ上で実行されます。
- 変換されたデータベースでは、以前のバージョンの 4D と同様、これらのコマンドはクライアントマシン上で実行されます。
- 変換されたデータベースで、特定の環境設定 (アプリケーション / 互換性ページ) を使用して、これらのコマンドの実行場所をグローバルに変更できます。

この実行場所の違いはアプリケーションパフォーマンスだけでなく (サーバ上での実行が通常早い)、プログラムにも影響します。実際、フォーミュラのコンポーネントの値は (特にメソッドで使用される変数)、実行コンテキストにより異なります。このセレクタを使用して、アプリケーションの動作を適応させることができます。

value 引数に 0 を渡すと、フォーミュラによるクエリコマンドの実行場所は、データベース設定に基づきます。4D v11 SQL で作成されたデータベースでは、これらのコマンドはサーバ上で実行されます。変換されたデータベースでは、データベースの環境設定に基づき、クライアントマシンまたはサーバマシン上で実行されます。value に 1 または 2 を渡すと、これらのコマンドは指定されたとおり、サーバマシンまたはクライアントマシンで実行されます。

- ▶ この例では、一時的にフォーミュラによるクエリをクライアントマシンで実行します：

```
curVal:= Get database parameter([table1];Query By Formula On Server)
  ` 現在の設定を格納
SET DATABASE PARAMETER([table1];Query By Formula On Server;2)
  ` クライアントマシンで実行するよう強制
QUERY BY FORMULA([table1];myformula)
SET DATABASE PARAMETER([table1];Query By Formula On Server;curVal)
  ` 設定を元に戻す
```

Selector = 47 (Order By Formula On Server)

- 値: 0 (データベース設定を使用する)、1 (サーバで実行) または 2 (クライアントで実行)
- 説明: パラメタで渡したテーブルに対する ORDER BY FORMULA コマンドの実行場所。

データベースをクライアント / サーバモードで実行しているとき、ORDER BY FORMULA コマンドはサーバ上あるいはクライアントマシン上で実行できます。このセレクタを使用して、このコマンドの実行場所を指定できます (サーバまたはクライアント)。詳細はセレクタ 46、Query By Formula On Server を参照してください。

Selector = 48 (Auto Synchro Resources Folder)

- 値: 0 (同期しない)、1 (自動同期)、または 2 (確認する)。
- 説明: コマンドを実行する 4D クライアントマシンとサーバの、Resources フォルダの動的同期モード。
サーバ上の Resources フォルダの内容が変更されたか、ユーザが同期をリクエストしたとき (例えばリソースエクスプローラで)、サーバは接続しているクライアントマシンにそれを通知します ([ページ 25](#)、"[クライアントマシンの通知](#)" の段落参照)。
クライアント側では、3 つの同期モードを利用できます。Auto synchro resources folder セレクタを使用して、カレントセッションのクライアントマシンで使用するモードを指定できます:
 - 0 (デフォルト値) = 動的同期を行わない (同期リクエストを無視する)
 - 1 = 自動で動的同期を行う
 - 2 = クライアントマシンにダイアログを表示し、同期の許可あるいは拒否を選択させる
 アプリケーションの環境設定を使用して、同期モードをグローバルに設定できます。詳細は [ページ 25](#)、"[クライアントマシンのアップデート](#)" の段落を参照してください。

Selector = 49 (QUERY BY FORMULA Joins)

- 値: 0 (データベース設定を使用)、1 (常に自動リレートを使用)、または 2 (可能であれば SQL JOIN を使用)。
- 説明: SQL JOIN の利用に関連する、QUERY BY FORMULA と QUERY SELECTION BY FORMULA コマンドの動作モード。
データベースが 4D v11 SQL のバージョン 11.2 以降で作成された場合、これらのコマンドは SQL の JOIN モデルを使用して実行されます ([ページ 53](#)、"[QUERY BY FORMULA, QUERY SELECTION BY FORMULA](#)" の段落参照)。このメカニズムを使用して、これらのテーブルが自動リレーションによって接続されていない場合でも (以前のバージョンの 4D では必要な条件でした)、他のテーブルに対して実行されたクエリの結果に基づき、テーブルのセレクションを変更できます。
QUERY BY FORMULA Joins セレクタを使用して、カレントプロセスのフォーミュラによるクエリコマンドの動作を指定できます:
 - ・ 0 (デフォルト値) = データベースの現在の設定を使用する。
 4D v11 SQL のバージョン 11.2 以降で作成されたデータベースでは、"SQL JOIN" は常にフォーミュラによるクエリで有効です。

変換されたデータベースでは、互換性のためデフォルトでこのメカニズムは無効になっていますが、環境設定を使用して実装することができます ([ページ 54](#)、"[新しい互換性環境設定](#)" の段落参照)。

- ・ 1 = 常に自動リレートを使用する (= 以前の 4D と同じ動作)。このモードでは、他のテーブルで実行されたクエリに基づきテーブルのセレクションを作成するには、リレーションが必要です。4D は "SQL JOIN" を使用しません。
- ・ 2 = 可能であれば SQL JOIN を使用する (= 4D v11 SQL のバージョン 11.2 以降で作成されたデータベースのデフォルト動作)。このモードでは、フォーミュラが適応する場合、4D は "SQL JOIN" をフォーミュラによるクエリに使用します (2 つの例外があります、[ページ 53](#)、"[QUERY BY FORMULA, QUERY SELECTION BY FORMULA](#)" の段落参照)。

システム環境

LOG EVENT

LOG EVENT ({outputType; }message{; importance})

引数	型	説明
outputType	整数	→ メッセージ出力タイプ
message	文字列	→ メッセージの内容
importance	整数	→ メッセージの重要度

LOG EVENT コマンドの能力が拡張されました。このコマンドを Mac OS 上でも使用することができ、新しいオプションの引数、outputType を使用してメッセージが使用する出力チャンネルを指定できます。

outputType には、"Log Events" テーマの以下の定数を渡すことができます：

定数	型	値
Into Windows Log events	倍長整数	0
Into debug message	倍長整数	1
Into 4D requests log	倍長整数	2
Into 4D commands log	倍長整数	3

注 コマンドの変更を反映させるため、"Windows Log Events" 定数テーマ名は "Log Events" に変更されました。

- Into Windows Log events
メッセージは Windows のログイベントに送信されます。Mac OS では、コマンドは何も行いません (以前のコマンドの動作に対応)。
- Into debug message
結果はプラットフォームにより異なります：
 - Mac OS: コマンドはコンソールにメッセージを送信します。

- Windows: コマンドはデバッグメッセージとしてメッセージを送信します。このメッセージを読めるようにするためには、Microsoft Visual Studio または Windows の DebugView ユーティリティが必要です (<http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx>)。
 - Into 4D requests log
4D のリクエストログファイルが有効になっていれば、コマンドはこのファイルにメッセージを記録します。
 - Into 4D commands log
4D のコマンドログファイルが有効になっていれば、コマンドはこのファイルにメッセージを記録します。
-
- 注 4D のログファイルは Logs フォルダに集められています。このフォルダはデータベースストラクチャファイルと同階層に作成されます。
-

リストボックス

GET LISTBOX ARRAYS

GET LISTBOX ARRAYS ({*; }object; arrColNames; arrHeaderNames; arrColVars; arrHeaderVars; arrVisible; arrStyles)

リストボックスが配列タイプの時、arrStyles の 4 番目のポイントとして、リストボックスの行表示を定義する配列を含むことができます。この配列はブールタイプです。

非表示行に関する詳細は、[ページ 16](#)、"[リストボックス](#)" の段落を参照してください。

セレクション

TRUNCATE TABLE

TRUNCATE TABLE {(!aTable)}

引数	型	説明
aTable	テーブル →	すべてのレコードを削除するテーブル、またはこの引数が省略された場合デフォルトテーブル

TRUNCATE TABLE コマンドは、aTable のすべてのレコードを素早く削除します。テーブルが既に空の場合、TRUNCATE TABLE は何も行いません。

このコマンドの効果は ALL RECORDS / DELETE SELECTION の連続した呼び出しと同様ですが、以下の点で異なります：

- トリガは呼び出されません。
- データ参照整合性はチェックされません。
- TRUNCATE TABLE を実行するプロセスはトランザクション中であってはなりません。もしトランザクション中の場合、コマンドは何も行わず、OK システム変数に 0 が設定されます。
- 1 つ以上のレコードが他のプロセスによりロックされていた場合、コマンドの実行に失敗します。エラーが生成され、OK システム変数に 0 が設定されます。LockedSet システムセットは作成されません。
- コマンドが呼びだされた後、カレントセクションもカレントレコードもありません。
- コマンドの実行に成功したか失敗したかにより、OK 変数に 1 または 0 が設定されます。
- ログファイルがある場合、操作は記録されます。

このため、TRUNCATE TABLE コマンドは注意の上使用しなければなりませんが、一時的なデータを素早く削除したい場合など特定のケースでは、とても利用価値があります。

注 このコマンドのコンセプトや機能は SQL の TRUNCATE (TABLE) のそれと同じです。

ピクチャ

BLOB TO PICTURE

BLOB TO PICTURE(*pictureBlob*; *picture*{; *codec*})

引数	型	説明
<i>pictureBlob</i>	BLOB	→ ピクチャが格納された BLOB
<i>picture</i>	ピクチャ	→ 4D ピクチャフィールドまたは変数
<i>codec</i>	文字列	→ ピクチャ codec ID

BLOB TO PICTURE コマンドは、オプションの 3 番目の引数 *codec* を受け入れるようになりました。BLOB をデコードする際に使用する *codec* を指定するために、この引数に文字列を渡すことができます。

codec 引数に 4D が認識できる *codec* を渡すと、それが BLOB に適用されます。PICTURE CODEC LIST コマンドを使用して利用可能なフォーマット

のリストを取得できます。

codec が BLOB を正しく認識すると、4D はピクチャを返し、OK 変数に 1 が設定されます。codec が BLOB を正しく認識できないと、4D は空のピクチャを返し、OK 変数に 0 が設定されます。

codec 引数に 4D が認識できない codec を渡すと、新しい codec が引数に渡された ID で動的に記録されます。4D は BLOB をカプセル化したピクチャを返し、OK 変数に 1 が設定されます。この場合、BLOB を取り出すには、同じ ID で PICTURE TO BLOB コマンドを使用しなければなりません。この特別なメカニズムは 2 つの特定のニーズのために使用できます：

- (ピクチャでない) BLOB をピクチャにカプセル化する。
- codec を使用せずにピクチャをロードする。

特にこれらのメカニズムの実装は、ピクチャ配列を使用して "BLOB 配列" を作成するために使用できます。このテクニックは注意して使用しなければなりません。配列は全体がメモリにロードされるため、大きなサイズの BLOB を使用すると、アプリケーションの動作に影響します。

注 VARIABLE TO BLOB コマンドで作成された BLOB は自動で管理されます。この BLOB には "署名" されているため、カプセル化するために codec を渡す必要はありません。この場合、BLOB を取り出すには、".4DVarBlob" を PICTURE TO BLOB コマンドの codec ID に渡します。

codec 引数を渡さない場合、以前のバージョンのように、4D はピクチャタイプの決定を試み、利用可能な codec を使用します。BLOB を認識できる codec がない場合、4D は空のピクチャを返し、OK 変数に 0 が設定されます。

READ PICTURE FILE

READ PICTURE FILE (fileName; picture{; *})

引数	型	説明
fileName	文字列	→ 読み込むファイルの名前またはパス名、または空の文字列
picture	ピクチャ	← ピクチャを受け取るフィールドまたは変数
*	*	→ 渡された場合 = すべてのタイプのファイルを受け入れる

READ PICTURE FILE コマンドはオプションの引数 * を受け入れるようになりました。この引数を渡すと、コマンドはすべてのタイプのファイルを受け入れます。これにより、適切な codec を持たずとも、ピクチャを扱うことができます。([ページ 51 のコマンド BLOB TO PICTURE](#) 参照)。

クエリ

QUERY SELECTION WITH ARRAY

QUERY SELECTION WITH ARRAY (targetField; array)

引数	型	説明
targetField	フィールド	→ 値を比較するために使用するフィールド
array	配列	→ 検索する値の配列

新しい QUERY SELECTION WITH ARRAY コマンドは、第一引数で渡されたテーブルのフィールドに対し、targetField の値が少なくとも配列の一つの要素値と一致するレコードを検索します。検索されたレコードは新しいカレントセレクションとなります。

QUERY SELECTION WITH ARRAY は QUERY WITH ARRAY と同様に動作します。これら 2 つのコマンドの異なる点はそのスコープです：

- QUERY WITH ARRAY は、targetField を含むテーブルのすべてのレコードを検索します。
- QUERY SELECTION WITH ARRAY は、targetField を含むテーブルのカレントセレクションのレコードのみを検索します。

詳細は QUERY WITH ARRAY コマンドの説明を参照してください。

QUERY BY FORMULA, QUERY SELECTION BY FORMULA

QUERY BY FORMULA(aTable{; queryFormula})

QUERY SELECTION BY FORMULA(aTable{; queryFormula})

QUERY BY FORMULA と QUERY SELECTION BY FORMULA コマンドは、SQL のように JOIN を実行します。つまり、[Table_A]field_X = [Table_B]field_Y のような queryFormula を使用する際、Table_A と Table_B の間に自動リレーションが設定されている必要はありません。

注 この新しいメカニズムは、既存のアプリケーションで非互換を発生させる可能性があるため、無効にすることができます。環境設定を使用してグローバルに（変換されたデータベースのみ）、あるいは [SET DATABASE PARAMETER](#), [Get database parameter](#) コマンドを使用してプロセスごとに設定できます。

"SQL JOIN" モードが有効な時でも以下のようなケースでは、QUERY BY FORMULA と QUERY SELECTION BY FORMULA はストラクチャエディタで設定された自動リレーションを使用します：

- queryFormula を { field ; comparator ; value } の形式に分解できない場合

- 同じテーブルの 2 つのフィールドが比較されているとき
- ▶ この例では、特定のフォーミュラによるクエリで、SQL JOIN を有効にします：

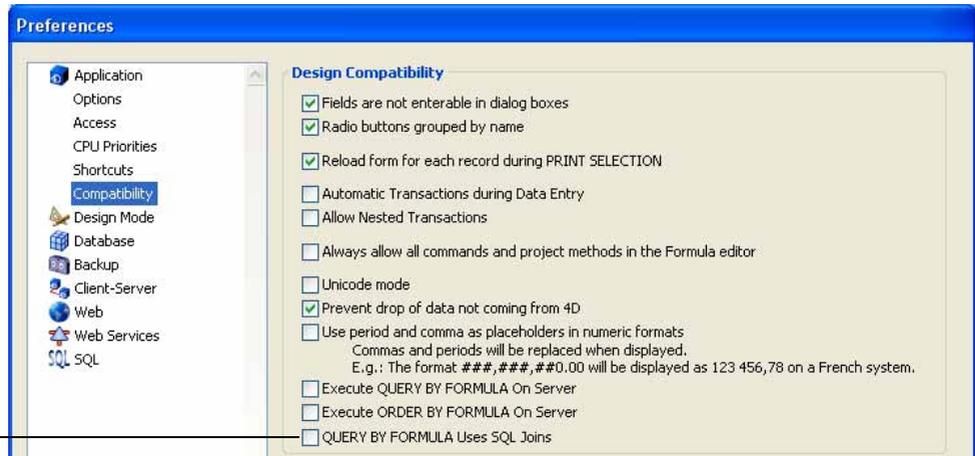
```

$currentval:= Get database parameter(QUERY BY FORMULA Joins)
  `SQL JOIN を有効にする
SET DATABASE PARAMETER(QUERY BY FORMULA Joins;2)
  `テーブルがリレートされていないすべての "ACME" 顧客を検索します
QUERY BY FORMULA([invoice_line]; [invoice_line]invoice_id =
[invoice]id & [invoice]client = "ACME")
  `クエリ前の設定に戻します
SET DATABASE PARAMETER(QUERY BY FORMULA Joins;$currentval)

```

新しい互換性環境設定

新しい QUERY BY FORMULA で SQL JOIN を使用する互換性環境設定を使用して、フォーミュラによるクエリ時の "SQL JOIN" の有効 / 無効を管理できます：



新しい環境設定

このオプションはバージョン 11.2 以降に変換されたデータベースでのみ表示されます。この設定はフォーミュラによるクエリの新しいメカニズムに 4D アプリケーションを徐々に適合させるために使用できます。

処理の互換性を保つため、デフォルトでこの設定は選択されていません。クエリによるフォーミュラコマンドの最適化の恩恵を受けるため、この設定を有効にし、データベースのコードを見直すことをお勧めします。

注 4D v11 SQL のバージョン 11.2 以降で新規に作成されたデータベースでは、このオプションは表示されず、このメカニズムは常に有効です。

Web サービス (クライアント)

4D は DIGEST モードでの Web サービスの認証をサポートするようになりました。これを行うために、Web サービス (クライアント) テーマの 2 つのコマンドが変更されました。

注 DIGEST 認証方法に関する詳細は、4D Language Reference マニュアルの "接続セキュリティ" を参照してください。

AUTHENTICATE WEB SERVICE

AUTHENTICATE WEB SERVICE (name; password{; authMethod})

引数	型	説明
name	文字列	→ ユーザ名
password	文字列	→ パスワード
authMethod	倍長整数	→ 認証モード 0= 指定しない, 1=BASIC, 2=DIGEST

AUTHENTICATE WEB SERVICE コマンドはオプションの 3 番目の引数、authMethod を受け入れるようになりました。この引数を使用して、次の CALL WEB SERVICE 呼び出しの際に使用する認証方法を指定できます。

authMethod には以下のいずれかの値を指定できます：

- 2 = DIGEST 認証を使用する
- 1 = BASIC 認証を使用する
- 0 (または引数が省略された) = 最も適切な方法を使用する

注 [SET WEB SERVICE OPTION](#) コマンドの 2 つの新しいオプションを使用して、DIGEST モードでの SOAP リクエストの動作をコントロールできます (認証ダイアログの表示と ID の保持)。

SET WEB SERVICE OPTION

SET WEB SERVICE OPTION(option; value)

SET WEB SERVICE OPTION コマンドは option 引数に 2 つの新しいセレクタを受け入れます：

定数	型	値
Web Service display auth dialog	倍長整数	4
Web Service reset auth settings	倍長整数	5

これらのセレクタは DIGEST モードでの Web サービスコマンドの認証設定を行うために使用します。

- option = Web Service display auth dialog
value = 0 (ダイアログを表示しない) または 1 (ダイアログを表示)
このオプションは、CALL WEB SERVICE コマンド実行時の認証ダイアログボックスの表示を管理します。デフォルトで、このコマンドはダイアログを表示しません。認証を行うには AUTHENTICATE WEB SERVICE コマンドを使用します。
しかし、ユーザに ID を入力させるためのダイアログを表示させたい場合、このオプションを使用し、value に 1 を渡します。ダイアログを表示させたくない場合は 0 を渡します。ダイアログは Web サービスに認証が必要な場合にのみ表示されます。
- option = Web Service reset auth settings
value = 0 (情報を消去しない) or 1 (情報を消去する)
このオプションを使用して、あとで再利用するために、4D に認証情報を記憶させるかさせないかを指定できます (ユーザ名、パスワード、メソッド等)。デフォルトでこの情報は、CALL WEB SERVICE を実行するたびに消去されます。value に 0 を渡すと情報が保持され、1 を渡すと消去されます。0 を渡した場合、情報はセッション中のみ保持されます。

Web サービス (サーバ)

4D は Web サービスのフレームワーク中で、XML ストラクチャの送信をサポートします。

SOAP DECLARATION SOAP DECLARATION(variable; type; input_output{; alias})

SOAP DECLARATION コマンドを使用して、"XML structure" と "DOM reference" タイプの変数を入出力両方で定義できるようになりました。このタイプのパラメタが指定されると、それらに処理やエンコードは行われません。データはそのまま送信されます。

これを行うために、2 つの新しい定数を type 引数に渡すことができます。これらの定数は "Web Services (Server)" テーマにあります：

定数	型	値
Is XML	倍長整数	36
Is DOM reference	倍長整数	37

- 出力パラメタ：
 - Is XML はパラメタに XML ストラクチャが含まれることを示します。
 - *Is DOM reference* はパラメタに XML ストラクチャの DOM 参照が含まれることを示します。この場合、SOAP メッセージに XML ストラクチャ

を挿入することは、DOM EXPORT TO VAR コマンドを実行することと同じです。

注 出力パラメタとして DOM 参照を使用する場合、グローバル参照の使用をお勧めします。これは例えば On Startup で作成され、アプリケーションが閉じられる際に破棄されません。実際、Web サービス自身で作成される DOM 参照を DOM CLOSE XML で閉じることはできません。そうでなければ、Web サービスは何も返せなくなってしまいます。そして Web サービスが複数呼び出されると、閉じられない DOM 参照が複数できてしまい、メモリが無駄に消費されてしまいます。

- 入力パラメタ：
 - Is XML はパラメタが SOAP クライアントから送信される XML 引数を受け取るべきであることを示します。
 - Is DOM reference はパラメタが SOAP クライアントから送信される、XML 引数に対応する XML ストラクチャの DOM 参照を受け取るべきであることを示します。

WSDL の変更

これらの XML ストラクチャは WSDL 内で、4D により、"anyType" タイプ (undetermined) として宣言されます。XML ストラクチャを明確に型付けしたい場合、WSDL ファイルを保存して、WSDL の <types> 部に適切なデータスキーマを追加しなければなりません。

- ▶ この例題では、異なるタイプの宣言による違いを示しています：

```
ALL RECORDS([Contact])
```

```

`Contacts のセレクションから XML ストラクチャを構築
`そして XML を BLOB に格納
C_BLOB(ws_vx_xmlBlob)
getContactsXML (->ws_vx_xmlBlob)
`XML ストラクチャをテキストに展開
C_TEXT(ws_vt_xml)
ws_vt_xml:=BLOB to text(ws_vx_xmlBlob;UTF8 text without length)
`XML ストラクチャの DOM 参照を取得
C_TEXT(ws_vt_xmlRef)
ws_vt_xmlRef:=DOM Parse XML variable (ws_vt_xml)

```

```

`さまざまな定義のテスト
SOAP DECLARATION(ws_vx_xmlBlob;Is BLOB ;SOAP Output ;
"contactListsX")
`4D が XML を Base64 に変換

```

```
SOAP DECLARATION(ws_vt_xml;Is Text;SOAP Output;"contactListsText")
```

`4D が XML をテキストに変換 (< > は実体参照になる)

`v 11.2

SOAP DECLARATION(ws_vt_xml;ls XML;SOAP Output;"xmlContacts")

`XML は XML テキストのまま渡される

SOAP DECLARATION(ws_vx_xmlBlob;ls XML;SOAP Output;
"blobContacts")

`XML は XML BLOB として渡される

SOAP DECLARATION(ws_vt_xmlRef;ls DOM reference;SOAP Output;
"contactByRef")

`XML は参照として渡される

レコードロード時トリガ

"レコードロード時"トリガは 4D でサポートされなくなりました。このトリガの動作は 4D v11 SQL に実装された内部的なメカニズムと互換性がありません。

変換されたデータベースでは、このトリガは呼び出されません。

4D バージョン 11.2 からは、ストラクチャのインスペクタパレットでこのトリガを選択できません。無効であることを示すために、On Loading Record Event 定数は _O_On Loading Record Event に名称が変更されました。

Web エリア

この新しいテーマのコマンドを使用して、4D v11 SQL バージョン 11.2 で加えられた新しい "Web エリア" オブジェクトを管理できます。"Web エリア" オブジェクトは [ページ 27](#)、"[Web エリア](#)" の段落で説明しています。

WA OPEN URL

WA OPEN URL({*;}object; url)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
url	文字列	→ Web エリアにロードする URL

WA OPEN URL コマンドは、url 引数に渡した URL を、* と object 引数で指定した Web エリアにロードします。

url に空の文字列を渡すと、WA OPEN URL コマンドは何も行わず、エラーも生成されません。Web エリアに空のページをロードするには、url に "about:blank" 文字列を渡します。

既存の [OPEN WEB URL](#) コマンドのように、WA OPEN URL はファイルを指定するための複数のシンタックスを url 引数に受け入れます：

- POSIX シンタックス : "file://c:/My%20File"
- システムシンタックス : "c:/MyFolder/MyFile" (Windows) または "MyDisk:MyFolder:MyFile" (Mac OS)

このコマンドは、Web エリアに関連付けられた "URL" 変数の値を更新することと、同じ効果があります。例として、エリアの URL 変数が MyWArea_url のとき：

```
MyWArea_url="http://www.4d.com/"
```

は以下と同じです：

```
WA OPEN URL(MyWArea;"http://www.4d.com/")
```

WA Get current URL

WA Get current URL ({*;}object) → 文字列

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
戻り値	文字列	← 現在 Web エリアにロードされている URL

WA Get current URL コマンドは、* と object 引数で指定した Web エリアに現在表示されているページの URL アドレスを返します。

現在の URL が利用できない場合、コマンドは空の文字列を返します。

Web ページが完全にロードされると、この関数から返される値は Web エリアに関連付けられた "URL" 変数のそれと同じです。ページがロード中の場合、2 つの値は異なります。関数は完全にロードされた URL を返し、変数はロード中の URL が格納されています。

- ▶ "www.apple.com" のページが表示されていて、"www.4d.com" ページをロード中の場合：

\$url:=WA Get current URL(MyWArea) ` 結果は "http://www.apple.com"
` 関連付けられた URL 変数は "http://www.4d.com"

WA OPEN BACK URL

WA OPEN BACK URL({*;}object)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)

WA OPEN BACK URL コマンドは、* と object 引数で指定した Web エリアに開かれた一連の URL 中、現在の URL のひとつ前の URL をロードしません。

前の URL がない場合、コマンドは何も行いません。前の URL が有効かどうかは、WA Back URL available コマンドで知ることができます。

WA Back URL available

WA Back URL available ({*;}object) → ブール

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
戻り値	ブール	← 開かれた一連の URL で、前の URL が有効なら True、そうでなければ False

WA Back URL available コマンドは、* と object 引数で指定した Web エリアに開かれた一連の URL で、前の URL が利用できるかどうかを知るために使用します。

コマンドは URL があれば True を、なければ False を返します。このコマンドは特に、ナビゲーションボタンを有効 / 無効にするために使用します。

WA OPEN FORWARD URL

WA OPEN FORWARD URL ({*;}object)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)

WA OPEN FORWARD URL コマンドは、* と object 引数で指定した Web エリアに開かれた一連の URL 中、現在の URL のひとつ次の URL をロードします。

次の URL がない場合、つまりユーザが前の URL に戻っていない場合、コマンドは何も行いません。次の URL が有効かどうかは、WA Forward URL available コマンドで知ることができます。

WA Forward URL available

WA Forward URL available ({*;}object) → ブール

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
戻り値	ブール	← 開かれた一連の URL で、次の URL が有効なら True、そうでなければ False

WA Forward URL available コマンドは、* と object 引数で指定した Web エリアに開かれた一連の URL で、次の URL が利用できるかどうかを知るために使用します。

コマンドは URL があれば True を、なければ False を返します。このコマンドは特に、ナビゲーションボタンを有効 / 無効にするために使用します。

WA REFRESH CURRENT URL

WA REFRESH CURRENT URL ({*;}object)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数

object	フォーム オブジェ クト	→	オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
--------	--------------------	---	--

WA REFRESH CURRENT URL コマンドは、* と object 引数で指定した Web エリアに現在表示されている URL を再読み込みします。

WA STOP LOADING URL

WA STOP LOADING URL ({*;}object)

引数	型		説明
*	*	→	渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→	オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)

WA STOP LOADING URL コマンドは、* と object 引数で指定した Web エリアの現在の URL のリソース読み込みを停止します。

WA Execute JavaScript

WA Execute JavaScript ({*;}object; jsCode) → 文字列

引数	型		説明
*	*	→	渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→	オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
jsCode	文字列	→	JavaScript コード
戻り値	文字列	←	実行結果

WA Execute JavaScript コマンドは、* と object 引数で指定した Web エリアで、jsCode に渡された JavaScript コードを実行します。

Mac OS では、コマンドは結果を返します。

Windows では、コマンドは空の文字列を返します。WA EXECUTE JAVASCRIPT FUNCTION コマンドを使用してください。

▶ 例:

```
$result:=WA Execute JavaScript(MyWArea;"history.back()")
```

WA EXECUTE JAVASCRIPT FUNCTION

WA EXECUTE JAVASCRIPT FUNCTION ({*;}object; jsFunction; result|*{*}; param1;...;paramN)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
jsFunction	文字列	→ 実行する JavaScript 関数の名前
result *	変数	→ 関数の結果 (結果を受け取る場合) ← または結果がない場合 *
param	文字列	→ Parameter(s) to pass to function

WA EXECUTE JAVASCRIPT FUNCTION コマンドは、* と object 引数で指定した Web エリアで、jsFunction に渡した JavaScript 関数を実行し、result 引数に結果を返します (オプション)。

関数が結果を返さない場合、result 引数に * を渡します。

param には関数の引数を含む文字列を 1 つ以上渡せます。

- ▶ 3 つの引数を使用して JavaScript 関数を呼び出す :

```
$JavaScriptFunction:="TheFunctionToBeExecuted"
$Param1:="10"
$Param2:="true"
$Param3:="1,000.2" `注 千区切りは "," で、小数点は "."
```

```
WA EXECUTE JAVASCRIPT FUNCTION(MyWArea; $JavaScriptFunction;
$Result; $Param1; $Param2; $Param3)
```

WA SET URL FILTERS

WA SET URL FILTERS ({*;}object; filtersArr; allowDenyArr)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
filtersArr	文字配列	→ フィルタ配列
allowDenyArr	ブール配列	→ 許可 - 拒否配列

WA SET URL FILTERS コマンドは、* と object 引数で指定した Web エリアで、1 つ以上のフィルタを設定するために使用します。

ユーザからリクエストされたページをロードする前に、4D はフィルタのリストを照会し、ターゲットの URL に接続可能かどうかを調べます。URL の判定は filtersArr と allowDenyArr 配列の内容に基づき行われます。

リクエストされた URL が許可されない場合、ページはロードされず、On URL Filtering フォームイベントが生成されます ([ページ 31](#)、" [フォームイベント](#) " の段落参照)。

filtersArr と allowDenyArr 配列は同期されていなければなりません。

- filtersArr 配列のそれぞれの要素には、フィルタする URL が含まれます。1 つ以上の文字を表すワイルドカードとして * を使用できます。
- allowDenyArr 配列のそれぞれ対応する要素には、URL を許可 (True) するか拒否 (False) するかを示すブール値が含まれます。

同じ URL が許可および拒否されているなど、設定レベルで矛盾がある場合、最後の設定が考慮されます。

フィルタを無効にするには、コマンドを呼び出す際に空の配列を渡すか、配列の最後の要素で、filtersArr 配列に "*" を、allowDenyArr 配列に True を渡します。

コマンドが実行されると、フィルタは Web エリアのプロパティとなります。filtersArr と allowDenyArr が削除されたり初期化されたりしても、コマンドが再実行されるまでフィルタは有効です。エリアで有効になっているフィルタを取得するには、[WA GET URL FILTERS](#) コマンドを使用しなければなりません。

重要: このコマンドによって実行されるフィルタは Web エリアに関連付けられた "URL" 変数にのみ適用されます (変数は通常入力可で、フォームに表示されます)。

フィルタは [WA OPEN URL](#) コマンドや他のナビゲーションコマンドには適用されません。

- ▶ .org, .net そして .fr Web サイトへのアクセスを禁止したい場合 :

```
ARRAY TEXT($filters;0)
ARRAY BOOLEAN($AllowDeny;0)
```

```
APPEND TO ARRAY($filters;"*.org")
APPEND TO ARRAY($AllowDeny;False)
APPEND TO ARRAY($filters;"*.net")
```

```

APPEND TO ARRAY($AllowDeny;False)
APPEND TO ARRAY($filters;"*.fr")
APPEND TO ARRAY($AllowDeny;False)
WA SET URL FILTERS(MyWArea;$filters;$AllowDeny)

```

- ▶ 日本のサイト以外へのアクセスを禁止したい場合 (.jp):

```

ARRAY TEXT($filters;0)
ARRAY BOOLEAN($AllowDeny;0)
APPEND TO ARRAY($filters;"*") `すべてを選択
APPEND TO ARRAY($AllowDeny;False) `すべて拒否
APPEND TO ARRAY($filters;"www.*.jp") `*.jpを選択
APPEND TO ARRAY($AllowDeny;True) `許可
WA SET URL FILTERS(MyWArea;$filters;$AllowDeny)

```

- ▶ 4D の Web サイトにのみアクセスを許可する場合 :

```

ARRAY TEXT($filters;0)
ARRAY BOOLEAN($AllowDeny;0)

APPEND TO ARRAY($filters;"*") `すべてを選択
APPEND TO ARRAY($AllowDeny;False) `すべて拒否
APPEND TO ARRAY($filters;"www.4D.*") `4d.fr, 4d.com... を選択
APPEND TO ARRAY($AllowDeny;True) `許可
WA SET URL FILTERS(MyWArea;$filters;$AllowDeny)

```

- ▶ ローカルのドキュメントにのみアクセスを許可 (C://doc フォルダ内):

```

ARRAY TEXT($filters;0)
ARRAY BOOLEAN($AllowDeny;0)

APPEND TO ARRAY($filters;"*") `すべてを選択
APPEND TO ARRAY($AllowDeny;False) `すべて拒否
APPEND TO ARRAY($filters;"file://C:/doc/*") `許可するファイルパスを選択
APPEND TO ARRAY($AllowDeny;True) `許可
WA SET URL FILTERS(MyWArea;$filters;$AllowDeny)

```

- ▶ 特定のキーワードを含むサイトを除いて許可する場合 :

```

ARRAY TEXT($filters;0)
ARRAY BOOLEAN($AllowDeny;0)

APPEND TO ARRAY($filters;"*")
APPEND TO ARRAY($AllowDeny;True) `すべて許可
APPEND TO ARRAY($filters;"*elcaro*") `elcaro を含むサイトを拒否
APPEND TO ARRAY($AllowDeny;False)
WA SET URL FILTERS(MyWArea;$filters;$AllowDeny)

```

- ▶ 特定の IP アドレスへのアクセスを拒否する場合：

```

ARRAY TEXT($filters;0)
ARRAY BOOLEAN($AllowDeny;0)

APPEND TO ARRAY($filters;"*") `すべてを選択
APPEND TO ARRAY($AllowDeny;True) `すべて許可
APPEND TO ARRAY(86.83.*") `86.83. で始まるアドレスを選択
APPEND TO ARRAY($AllowDeny;False) `拒否
APPEND TO ARRAY(86.1*") `86.1 で始まるアドレスを選択 (86.10, 86.135
etc.)
APPEND TO ARRAY($AllowDeny;False) `拒否
WA SET URL FILTERS(MyWArea;$filters;$AllowDeny)

(ドメインの IP アドレスは変わることがあります。)

```

See Also: [WA GET URL FILTERS](#)

WA GET URL FILTERS WA GET URL FILTERS({*;}object; filtersArr; allowDenyArr)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
filtersArr	文字配列	← フィルタ配列
allowDenyArr	ブール配 列	← 許可 - 拒否配列

WA GET URL FILTERS は、filtersArr と allowDenyArr 配列に、* と object 引数で指定した Web エリアで有効なフィルタを返します。フィルタが有効でない場合、空の配列が返されます。

フィルタは WA SET URL FILTERS コマンドでインストールされます。セッション中に配列が再初期化されても、WA GET URL FILTERS コマンドを使用すれば現在の設定を知ることができます。

See Also: [WA SET URL FILTERS](#)

WA SET EXTERNAL LINKS FILTERS WA SET EXTERNAL LINKS FILTERS({*;}object; filtersArr; allowDenyArr)

引数	型	説明
----	---	----

*	*	→	渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→	オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
filtersArr	文字配列	→	フィルタ配列
allowDenyArr	ブール配 列	→	許可 - 拒否配列

WA SET EXTERNAL LINKS FILTERS コマンドを使用して、* と object 引数で指定した Web エリアの外部リンクフィルタを設定できます。外部リンクフィルタは、リンクを使用して現在のページに関連付けられている URL を Web エリアで開くか、マシンのデフォルト Web ブラウザで開くかどうか決定するために使用されます。

ユーザが現在のページでリンクをクリックすると、4D は外部リンクフィルタのリストを照会し、リクエストされた URL をマシンのブラウザで開くかチェックします。開く場合、URL に対応するページが Web ブラウザで開かれ、On Open External Link フォームイベントが生成されます ([ページ 31](#)、" [フォームイベント](#) " の段落参照)。そうでなければ (デフォルト動作)、URL に対応するページは Web エリア内に表示されます。URL の判定は filtersArr と allowDenyArr 配列の内容に基づき行われます。

filtersArr と allowDenyArr 配列は同期されていなければなりません。

- filtersArr 配列のそれぞれの要素には、フィルタする URL が含まれます。1 つ以上の文字を表すワイルドカードとして * を使用できます。
- allowDenyArr 配列のそれぞれ対応する要素には、URL を Web エリアで表示する (True) か Web ブラウザで表示する (False) かを示すブール値が含まれます。

同じ URL が許可および拒否されているなど、設定レベルで矛盾がある場合、最後の設定が考慮されます。

フィルタを無効にするには、コマンドを呼び出す際に空の配列を渡すか、配列の最後の要素で、filtersArr 配列に "*" を、allowDenyArr 配列に True を渡します。

重要 : [WA SET URL FILTERS](#) コマンドで設定されたフィルタが、WA SET EXTERNAL LINKS FILTERS コマンドよりも前に評価されます。つまり URL が [WA SET URL FILTERS](#) コマンドフィルタの設定により拒否されると、WA SET EXTERNAL LINKS FILTERS コマンドで受け入れていても、その URL をブラウザで開くことはできません ([例 2](#) 参照)。

- ▶ この例はサイトを外部ブラウザで開きます :

```

ARRAY STRING(0;$filters;0)
ARRAY BOOLEAN($AllowDeny;0)

APPEND TO ARRAY($filters;"*www.google.*") `google" を選択
APPEND TO ARRAY($AllowDeny;False)
  `False: このリンクは外部ブラウザで開く
APPEND TO ARRAY($filters;"*www.apple.*")
APPEND TO ARRAY($AllowDeny;False)
  `False: このリンクは外部ブラウザで開く
WA SET EXTERNAL LINKS FILTERS(MyWArea;$filters;$AllowDeny)

```

- ▶ この例はサイトと外部リンク両方のフィルタを使用します :

```

ARRAY STRING(0;$filters;0)
ARRAY BOOLEAN($AllowDeny;0)
APPEND TO ARRAY($filters;"*www.google.*") `google" を選択
APPEND TO ARRAY($AllowDeny;False) `このリンクを拒否
WA SET URL FILTERS(MyWArea;$filters;$AllowDeny)

ARRAY STRING(0;$filters;0)
ARRAY BOOLEAN($AllowDeny;0)
APPEND TO ARRAY($filters;"*www.google.*") `google" を選択
APPEND TO ARRAY($AllowDeny;False)
  `False: このリンクは外部ブラウザで開かれるべきだが、この設定は
  `URL フィルタによりブロックされ、無効となる。
WA SET EXTERNAL LINKS FILTERS(MyWArea;$filters;$AllowDeny)

```

WA GET EXTERNAL LINKS FILTERS

WA GET EXTERNAL LINKS FILTERS({*;}object; filtersArr; allowDenyArr)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
filtersArr	文字配列	← フィルタ配列
allowDenyArr	ブール配 列	← 許可 - 拒否配列

WA GET EXTERNAL LINKS FILTERS コマンドは、filtersArr と allowDenyArr 配列に、* と object 引数で指定した Web エリアの外部リンクフィルタを返します。フィルタが有効でない場合、空の配列が返されます。

フィルタは [WA SET EXTERNAL LINKS FILTERS](#) コマンドでインストールされます。セッション中に配列が再初期化されても、WA GET EXTERNAL LINKS FILTERS コマンドを使用すれば現在の設定を取得できます。

See Also: [WA SET EXTERNAL LINKS FILTERS](#)

WA GET LAST URL ERROR

WA GET LAST URL ERROR ({*};object; url; description; errorCode)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
url	文字列	← エラー元の URL
description	文字列	← エラーの説明 (Mac OS)
errorCode	倍長整数	← エラーコード

WA GET LAST URL ERROR コマンドを使用して、* と object 引数で指定した Web エリアで発生した最後のエラーに関する情報のいくつかの項目を取得できます。

この情報は 3 つの変数に返されます：

- url: エラーを発生させた URL
- description (Mac OS のみ): エラーについての説明 (利用可能な場合)。エラーにテキストが関連づけられていない場合、空の文字列が返されます。Windows では、常に空の文字列が返されます。
- errorCode: エラーコード
 - コードが 400 以上の場合、それは HTTP プロトコル関連のエラーです。このタイプのエラーに関する詳細情報は、以下のアドレスを参照してください：
<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
 - 上記以外の場合、WebKit (Mac OS) または ActiveX (Windows) から返されたエラーです。

発生したエラーの原因を知るためには、On URL Loading Error フォームイベントのフレームワークでこのコマンドを呼び出すことをお勧めします。詳細な情報については [ページ 31](#)、"[フォームイベント](#)" の段落を参照してください。

WA Get last filtered URL

WA Get last filtered URL ({*;}object) → 文字列

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
戻り値	文字列	← 最後にフィルタされた URL

WA Get last filtered URL コマンドは、* と object 引数で指定した Web エリアで、最後にフィルタされた URL を返します。

URL は以下のいずれかの理由でフィルタされることがあります：

- URL がフィルタにより拒否された ([WA SET URL FILTERS](#) コマンド)。
- デフォルトブラウザでリンクが開かれる ([WA SET EXTERNAL LINKS FILTERS](#) コマンド)。
- URL がポップアップウィンドウを開こうとしている。

フィルタされた URL を知るためには、On URL Filtering、On Open External Link、そして On Window Opening Denied フォームイベントのコンテキストでこのコマンドを呼び出すことをお勧めします。詳細な情報については[ページ 31](#)、"[フォームイベント](#)" の段落を参照してください。

WA GET URL HISTORY

WA GET URL HISTORY ({*;}object; urlsArr{; direction{; titlesArr{}})

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
urlsArr	文字配列	← 訪問した URL の配列
direction	整数	→ 0 または省略 = 前の URL リスト 1 = 次の URL リスト
titlesArr	文字配列	← ウィンドウタイトルのリスト

WA GET URL HISTORY コマンドは、* と object 引数で指定した Web エリアのセッション中に訪問した URL を含む、1 つまたは 2 つの配列を返し

ます。このコマンドを使用して、カスタムのナビゲーションシステムを作成できます。

提供される情報はセッションに限定されます。言い換えればナビゲーションは同じ Web エリアで、フォームが閉じられない間実行されます。

urlsArr 配列には、訪問した URL が返されます。direction 引数により (渡されれば)、配列には前の URL (デフォルト動作)、または次の URL が返されます。これらのリストは、ブラウザ標準の次のページや前のページボタンの内容に対応します。

URL は時間順に分類されます。

direction には取得するリストを指定する値を渡します。"Web Area" テーマの以下の定数を使用できます：

定数	型	値
wa previous URLs	倍長整数	0
wa next URLs	倍長整数	1

direction 引数を省略すると、0 が使用されます。

titlesArr 引数を渡すと、URL に関連付けられたウィンドウの名前が返されます。この配列は urlsArr 配列と同期しています。

WA Create URL history menu

WA Create URL history menu ({*;}object{; direction}) → MenuRef

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
direction	整数	→ 0 または省略 = 戻る URL リスト 1 = 進む URL リスト
戻り値	MenuRef	← メニュー参照

WA Create URL history menu コマンドは、* と object 引数で指定した Web エリアのセッション中に訪問した URL 間をナビゲートするメニューを作成します。このコマンドを使用して、カスタムのナビゲーションシステムを作成できます。

提供される情報はセッションに限定されます。言い換えればナビゲーションは同じ Web エリアで、フォームが閉じられない間実行されます。

Pdirection には取得するリストを指定する値を渡します。"Web Area" テーマの以下の定数を使用できます：

定数	型	値
wa previous URLs	倍長整数	0
wa next URLs	倍長整数	1

direction 引数を省略すると、0 が使用されます。

メニューが生成されたら、4D の Dynamic pop up menu コマンドを使用してそれを表示することができ、また 4D のメニュー管理コマンドを使用することもできます。Dynamic pop up menu コマンドから返される ItemRef 参照 (文字型) には、訪問したページの URL が含まれます (例参照)。

メニューを使用しなくなったら、RELEASE MENU コマンドを呼び出して URL 履歴メニューを削除します。

- ▶ 以下のコードを、ポップアップメニューを持つ "戻る" 3D ボタンに関連付けることができます：

Case of

 `シングルクリック

 :(Form event=On clicked)

 WA OPEN BACK URL (WA_area)

 `矢印をクリック -> ポップアップ表示

 :(Form event=On Arrow Click)

 `前の履歴メニューを作成

 \$Menu:=WA Create URL history menu (WA_area;wa previous URLs)

 `ポップアップにこのメニューを表示

 \$URL:=Dynamic pop up menu (\$Menu)

 `項目が選択されれば

 If (\$URL#"")

 `Web ページを開く

 WA OPEN URL(WA_area;\$URL)

 End if

 `メニューを削除してメモリを解放

 RELEASE MENU(\$Menu)

End case

See Also: Dynamic pop up menu, RELEASE MENU

WA Get page title

WA Get page title ({*;}object) → String

引数	型	説明
----	---	----

*	*	→	渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→	オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
戻り値	文字列	←	現在のページのタイトル

WA Get page title コマンドは、* と object 引数で指定した Web エリアの現在のページまたは表示されているページのタイトルを返します。タイトルは HTML の "title" タグに対応します。

現在の URL でタイトルが利用できない場合、このコマンドは空の文字列を返します。

WA SET PAGE CONTENT

WA SET PAGE CONTENT({*;}object; content; baseURL)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
content	文字列	→ HTML ソースコード
baseURL	文字列	→ 相対参照に使用する URL (Mac OS)

WA SET PAGE CONTENT コマンドは、* と object 引数で指定した Web エリアに表示されているページを、content 引数で渡された HTML コードで置き換えます。

Mac OS では、baseURL 引数を使用して、ページ中に存在する相対パスの前に追加するベース URL を指定できます。

Windows では、この引数は効果がなく、ベース URL は指定されません。このプラットフォームでは相対参照を使用できません。

注 Windows では、このコマンドが呼ばれる前に、ページが既に Web エリアにロードされていなければなりません。必要であれば "about:blank" URL を使用してブランクページをロードできます。

- ▶ "Hello world!" を表示して、ベース URL を "file:/// " にします (Mac OS のみ):

```
WA SET PAGE CONTENT(MyWArea;"<html><body><h1>Hello
```

```
World!</h1></body></html>";"file:///")
```

WA Get page content WA Get page content({*;}object) → 文字列

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
戻り値	文字列	← HTML ソースコード

WA Get page content コマンドは、* と object 引数で指定した Web エリアの現在のページまたは表示されているページの HTML コードを返します。

現在のページの内容を利用できない場合、このコマンドは空の文字列を返します。

WA SET PAGE TEXT LARGER

WA SET PAGE TEXT LARGER({*;}object)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)

WA SET PAGE TEXT LARGER コマンドは、* と object 引数で指定した Web エリアに表示されているテキストのサイズを大きくします。

Mac OS では、このコマンドのスコープは 4D セッションとなります。このコマンドにより実行される設定は、4D アプリケーション終了後は引き継がれません。

Windows では、このコマンドのスコープはグローバルです。4D アプリケーション終了後も設定が引き継がれます。

WA SET PAGE TEXT SMALLER

WA SET PAGE TEXT SMALLER({*;}object)

引数	型	説明
----	---	----

*	*	→	渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→	オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)

WA SET PAGE TEXT SMALLER コマンドは、* と object 引数で指定した Web エリアに表示されているテキストのサイズを小さくします。

Mac OS では、このコマンドのスコープは 4D セッションとなります。このコマンドにより実行される設定は、4D アプリケーション終了後は引き継がれません。

Windows では、このコマンドのスコープはグローバルです。4D アプリケーション終了後も設定が引き継がれます。

WA SET PREFERENCE WA SET PREFERENCE({*};object; selector; value)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェ クト	→ オブジェクト名 (* が渡されたとき) または 変数 (* が省略されたとき)
selector	倍長整数	→ 変更する環境設定
value	Boolean	→ 環境設定の値 (True = 許可, False = 不許可)

WA SET PREFERENCE コマンドは、* と object 引数で指定した Web エリアの環境を設定するために使用します。

selector 引数に変更する環境設定項目を渡し、value 引数に値を渡します。selector には、"Web Area" テーマの以下の定数を使用できます：

定数	型	値
wa enable Java applets	倍長整数	0
wa enable JavaScript	倍長整数	1
wa enable plugins	倍長整数	2
wa enable contextual menu	倍長整数	3

それぞれの環境設定に対し、有効にするには True を、無効にするには False を渡します。

それぞれのセレクタの意味は以下のとおりです：

- wa enable Java applets: Web エリアでの Java アプレットの実行を許可するために使用します。
- wa enable JavaScript: Web エリアでの JavaScript の実行を許可するために使用します。
- wa enable plugins: Web エリアでのプラグインのインストールを許可するために使用します。
- wa enable contextual menu: Web エリアでの標準のコンテキストメニューの表示を許可するために使用します。

See Also: [WA GET PREFERENCE](#)

WA GET PREFERENCE WA GET PREFERENCE({*};object; selector; value)

引数	型	説明
*	*	→ 渡した場合、object はオブジェクト名 (文字列) 省略した場合、object は変数
object	フォーム オブジェクト	→ オブジェクト名 (* が渡されたとき) または変数 (* が省略されたとき)
selector	倍長整数	→ 取得する環境設定
value	変数	← 環境設定の現在の値

WA GET PREFERENCE コマンドは、* と object 引数で指定した Web エリアの現在の環境設定値を取得するために使用します。

selector 引数に値を取得する環境設定項目を渡します。"Web Area" テーマの以下の定数を使用できます：

定数	型	値
wa enable Java applets	倍長整数	0
wa enable JavaScript	倍長整数	1
wa enable plugins	倍長整数	2
wa enable contextual menu	倍長整数	3

これらの環境設定についての詳細は、[ページ 75 のコマンド WA SET PREFERENCE](#) の説明を参照してください。

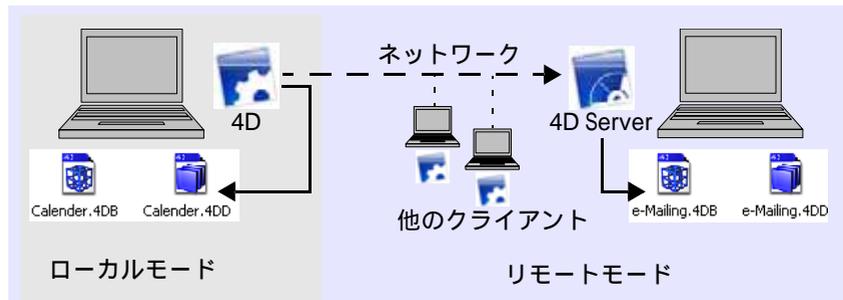
4

4D Server v11 SQL

4D Server v11 SQL によろこそ。この新しいバージョンの 4D Server では多くの重要な変更が施され、4D v11 SQL の製品ラインで提供されるパワーにを完全にご利用いただけるようになりました。

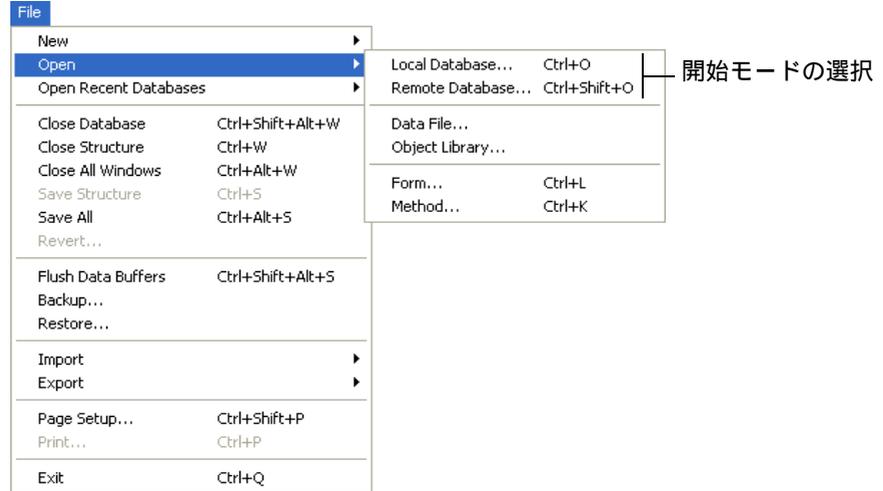
アーキテクチャ

バージョン 11 では、4D のクライアント / サーバアーキテクチャがシンプルになりました。特に 4D Client が 4D シングルユーザ (4D Developer) に統合されました。これからはローカルデータベースまたは 4D Server で実行されるリモートデータベースに接続する際に、1つのアプリケーション 4D を使用します。前者のケースではアプリケーションはローカルモード (シングルユーザ) で動作します。後者のケースでは、クライアント / サーバモードで動作します。



それぞれのモードを使用するには、適切なライセンスが必要です。

開くデータベースを選択する際、開くデータベースのタイプに基づき、使用するモードを選択します：



- 開く / ローカルデータベース：標準のドキュメントを開くダイアログが表示され、4D ストラクチャファイルを選択できます。
- 開く / リモートデータベース：4D Server への接続ダイアログが表示され、公開されたデータベースを選択できます。

注：4D Server データベースに直接接続するために使用されていた .pth ファイルは、XML ファイル (拡張子 .4DLink) に置き換えられました。詳細は [ページ 35](#)、" [データベース接続の保存](#) " の段落を参照してください。

リモートフォルダの管理

Resources と Extras フォルダ

サーバマシンとクライアントマシン間のカスタムデータ (ピクチャ、ファイル、サブフォルダ等) の転送は、データベースストラクチャと同階層にある Resources を通して行います。このフォルダの内容は、クライアントマシンが接続する際に自動で更新されます。Resources に関連付けられたすべての参照メカニズム (.lproj フォルダ、XLIFF、ピクチャなど) はクライアント / サーバモードでサポートされています。さらに、4D v11 SQL はこのフォルダを動的に管理するために使用できるさまざまなツール、特にリソースエクスプローラを提供します ([ページ 17](#) " [リソースエクスプローラ](#) " の段落を参照) 。

したがって、バージョン 11.2 の 4D v11 SQL から、Extras フォルダの利用は推奨されません。このフォルダには Resources フォルダで提供され

ているようなメカニズムの利点はありません。今バージョンから、Extras フォルダの代わり Resources フォルダを使用する必要があります。ただし Extras フォルダは既存のアプリケーションとの互換性を保つため、4D Server によりサポートされます。

注： 4D v11 SQL でリモートの 4D マシン (クライアント) にダウンロードされる内部フォルダの場所が変更されました。詳細は [ページ 44 "Get 4D folder" の段落](#) を参照してください。

Components フォルダ

クライアント / サーバモードで、コンポーネントの開発とインストールはローカルモードと同じ方法で行います。例外は、クライアント / サーバモードでは、データベースの Components フォルダ内部にコンポーネントへのショートカット (Windows) またはエイリアス (Mac OS) を使用できません。

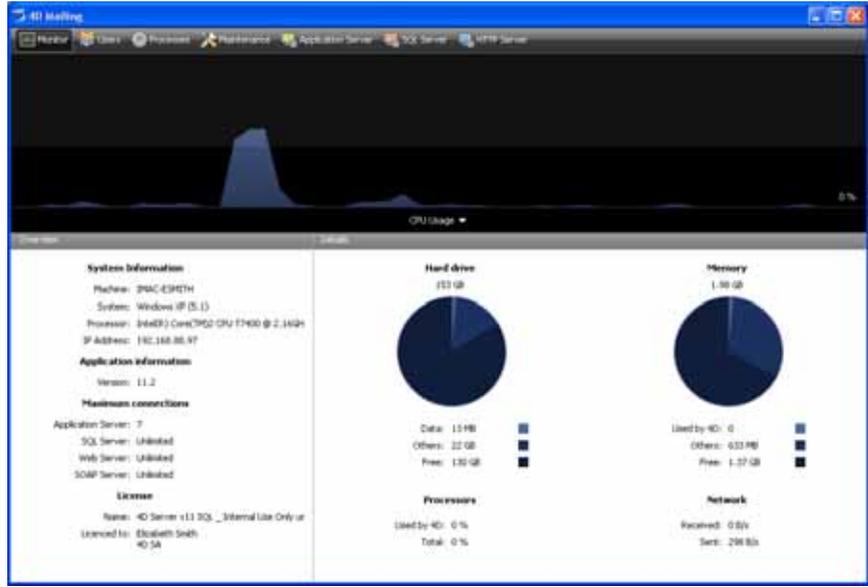
クライアント側では、Components フォルダは他のストラクチャ要素と同じ場所にダウンロードされます：

システム	クライアントマシン上での Components フォルダの場所
Mac OS	{Disk}:/users/{user_name}/library/Caches/ 4D:DatabaseName_Address:Components
Windows XP	{Disk}:/Documents and Settings/{user_name}/Local Settings/Application Data/DatabaseName_Address/ Components
Windows Vista	{Disk}:/Users/{user_name}/AppData/Local/4D/ DatabaseName_Address/Components

Components フォルダの元の階層構造がそれぞれのクライアントマシン上で再構築されます。

新しいサーバ管理ウィンドウ

4D Server v11 により包括的でユーザフレンドリな管理ウィンドウが実装されました：



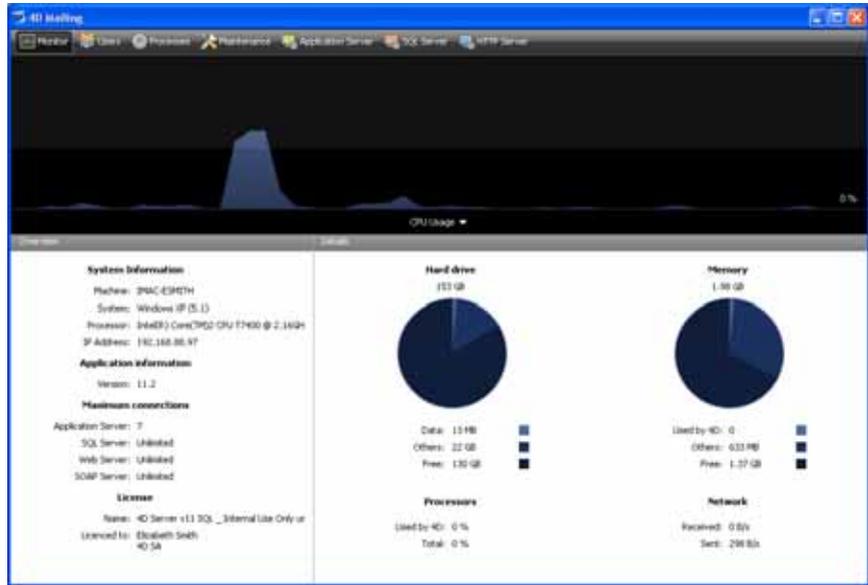
このウィンドウは公開されたデータベースのさまざまな分析やコントロールツールを提供します。ウィンドウは 8 のページで構成され、上部のボタンから各ページにアクセスできます：



それぞれのページについて以下説明します。

モニタ

モニタページにはデータベースの利用状況のほか、システムや 4D Server アプリケーションの情報が表示されます：



注：Windows では、表示される情報はセッションを開いたユーザの権限に関連します。詳細は [ページ 83](#)、" [モニタ情報の表示 \(Windows\)](#)" の段落を参照してください。

グラフィックエリア

グラフィックエリアにはリアルタイムで CPU 利用率、ネットワークトラフィック、またはメモリの変化が表示されます。ウィンドウの中央にあるメニューから表示させる内容を選択します：

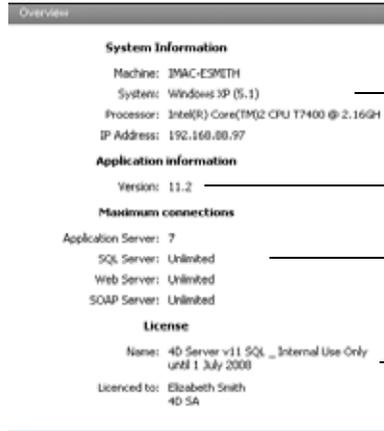


- CPU 利用率：すべてのアプリケーションによるマシンの全体的な CPU 利用率。
この利用率中 4D Server の特定の部分については、" プロセッサ " 情報エリアで提供されます。
- ネットワーク：4D Server が秒あたりに受信したバイト数。
4D Server が送信したバイト数は " ネットワーク " 情報エリアで提供されます。

- **メモリ** : 4D Server が使用する、マシンの RAM の量。メモリの利用に関するより詳細な情報は "メモリ" 情報エリアで提供されます。

概要エリア

"概要" エリアではシステム、アプリケーション、そして 4D Server マシンにインストールされたライセンスに関する様々な情報が提供されます。



—— コンピュータ、システム、そしてサーバの IP アドレス

—— 4D Server の内部バージョン

—— サーバのタイプにより許可される同時接続数

—— ライセンスの説明

詳細エリア

"詳細" エリアはグラフィックエリアで表示されていた情報の一部と、追加の情報を提供します。



- **ハードディスク** : ハードディスク全体およびデータベースデータ (データファイルとインデックスファイル) の使用スペース、他のファイルの使用スペース、空きスペースなどを表示します。
- **メモリ** : マシンにインストールされた RAM メモリ、4D Server の使用量、他のアプリケーションの使用量、そして空き容量。

4D Server が使用するメモリはグラフィックエリアにも動的に表示されます。

- プロセッサ : 4D Server と他のアプリケーションによる、プロセッサの使用率。この使用率は定期的に再計算されます。
4D Server による使用率は、グラフィックエリアにも動的に表示されます。
- ネットワーク : ネットワークから 4D Server が受信したその瞬間のバイト数、およびアプリケーションが送信したバイト数。この値は定期的に再計算されます。
4D Server が受信したバイト数は、グラフィックエリアにも動的に表示されます。

モニタ情報の表示 (Windows)

Windows では、モニタページに表示されるいくつかのシステム情報は、Windows の "Performance Analyzer" ツールを使用して取得しています。これらのツールは、4D Server を起動したセッションを開いたユーザが、必要な認証を得られている場合のみアクセスできます。このユーザは以下のいずれかでなければなりません :

- "Administrators" グループに属する
- Windows Vista: "Power Users" に属する (管理者でないユーザの場合)
- ▶ Windows Vista で管理者でないユーザを "Power Users" グループに入れるには :

注 : この操作を行うには、管理者アカウントを使用しなければなりません。

- 1 コントロールパネルから、" ユーザー アカウント " パネルを開く。
- 2 " 詳細 " タブをクリックし、" 詳細ユーザ管理 " エリアの " 詳細 " ボタンをクリックする
"Local Users and Groups" アプリケーションが実行されます。
- 3 "Groups" フォルダをダブルクリックする。
- 4 "Power Users" グループをダブルクリックする。
"Power Users Properties" という名称のウィンドウが表示されます。
- 5 追加... ボタンをクリックして、ユーザを追加する。
" 選択するオブジェクト名を入力 " テキストエリアに、許可したいユーザ名を入力する。
- 6 OK をクリックする (2 回)。
- 7 "Local Users and Groups" アプリケーションと " ユーザアカウント " を閉じる。

ユーザ

ユーザ ページには、データベースに接続したユーザが表示されます：

4D User	Machine name	Session name	IP Address	Login date	CPU Time	Activity
Super_Administrator	2PAC-179114	Admin	192.168.88.87	4/10/2008 13:36	00:00:03	1%
Super_Administrator	2PAC-AD324621	Arnaud Schmitt	192.168.88.105	4/10/2008 13:34	00:00:03	1%

Users (11) ボタンには、括弧内にデータベースに接続中のユーザ数が表示されます。(この番号は、ウィンドウに適用される表示フィルタでは考慮されません。)

このページには検索を行うためのエリアや、コントロールボタンもあります。

ヘッダエリアをドラッグ&ドロップして、列の順番を入れ替えることができます。またヘッダをクリックすると、リストの値が並び替えられます：

IP Address — クリックすると昇順 / 降順が切り替えられます。

ユーザのリスト

データベースに接続したユーザごとに、リストには以下の情報が表示されます：

- システム：クライアントマシンのシステム (Mac OS / Windows)。
- 4D ユーザ：4D ユーザ名。パスワードシステムが有効になっていない場合、すべてのユーザは "Designer" となります。
- マシン名：クライアントマシンの名前。
- セッション名：クライアントマシン上で開かれたセッション名。
- IP アドレス：クライアントマシンの IP アドレス。

- ログイン日 : クライアントマシンが接続した日付と時刻。
- CPU Time: 接続してからこのユーザが消費した CPU の時間
- Activity: 4D Server がこのユーザのために使用する時間の率 (動的表示)。

検索 / フィルタエリア

User;Machine;Session...

この機能を使用して、検索エリアに入力されたテキストに対応する行だけをリストに表示させ、行数を減らすことができます。エリアには、どの列に対して検索 / フィルタが実行されるかが表示されています。ユーザページでは、4D ユーザ、マシン名、そしてセッション名です。

リストはエリアにテキストが入力されると、リアルタイムで更新されます。

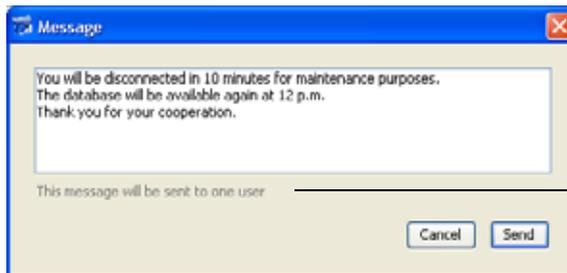
値をセミコロンで区切ることで、一つ以上の値を使用して検索を行うことができます。この場合 OR タイプの演算が行われます。

例えば、"John;Mary;Peter" と入力すると、John または Mary または Peter が上記の列にある行のみが表示されます。

管理ボタン

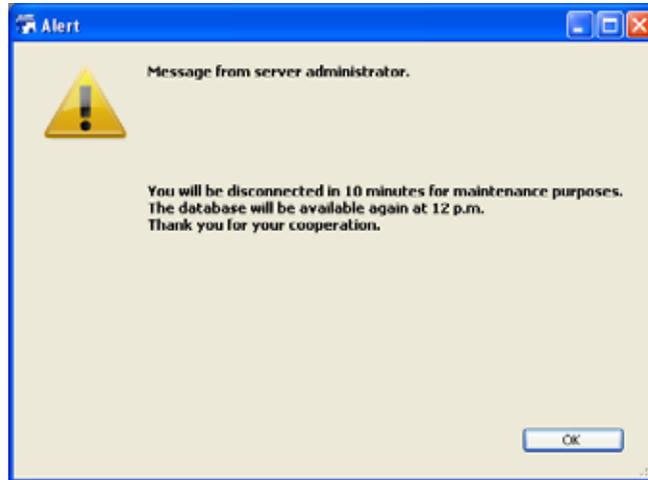
このページには 3 つのコントロールボタンがあります。これらのボタンは最低 1 つの行が選択されているときに有効になります。Shift キーを押しながらクリックして連続した行を、あるいは Ctrl (Windows) / Command (Mac OS) キーを押しながらクリックして連続しない行を複数選択できます。

- メッセージ送信
このボタンを使用して、ウィンドウで選択した 4D ユーザにメッセージを送信できます。ユーザが選択されていないと、ボタンを使用できません。ボタンをクリックするとダイアログボックスが表示され、メッセージを入力できます。ダイアログにはメッセージを受信するユーザの数が表示されます：



選択されたユーザ数

このメッセージはクライアントマシン上で、警告メッセージとして表示されます：



- プロセス表示
このボタンをクリックすると、選択されたユーザのプロセスを直接、管理ウィンドウのプロセス ページに表示させることができます。ボタンをクリックすると、4D Server はプロセスページに移動し、このページの検索 / フィルタエリアに選択されたユーザ名を入力します。詳細はこのページの説明を参照してください。
- 接続解除
このボタンは、選択したユーザの接続を強制的に解除するために使用します。
kのボタンをクリックすると警告ダイアログが表示され、接続解除を実行するかキャンセルするか選択できます。

注： 確認ダイアログを表示させずに選択したユーザの接続を解除できます。これを行うには、Alt (Windows) または Option (Mac OS) を押しながら、接続解除ボタンをクリックします。

プロセス

プロセス ページには実行中のプロセスが表示されます：

Process name	Session	Type	Num	State	CPU Time	Activity
4D4D Flush		SQL Server	0	Running	00:00:00	0.00 %
4D4D Index builder		4D4D Server	0	Running	00:00:00	0.00 %
4D4D Server		SQL Server	0	Running	00:00:00	0.00 %
EXCLUSIVE pool worker 0		SQL Server	0	Running	00:00:00	0.00 %
EXCLUSIVE pool worker 1		SQL Server	0	Running	00:00:00	0.00 %
Shared Load Balance		SQL Server	0	Running	00:00:00	0.00 %
SHARED pool worker 0		SQL Server	0	Running	00:00:00	0.00 %
SHARED pool worker 1		SQL Server	0	Running	00:00:00	0.00 %
SQL Net Session Manager Thread		SQL Server	0	Running	00:00:00	0.00 %
Client Manager		Application server	1	Waiting for Req	00:00:00	0.00 %
Internal Bridge Process		Application server	4	Waiting for Req	00:00:01	0.00 %
Internal Error Process		Application server	2	Executing	00:00:01	0.00 %
User Interface		Application server	1	Waiting for event	00:00:01	1.00 %

ボタンには、括弧内にデータベースで実行中のプロセス数が表示されます。(この番号は、ウィンドウに適用される表示フィルタやグループ毎にプロセスを表示オプションのステータスでは考慮されません。)

列ヘッダをドラッグ&ドロップして、列の順番を入れ替えることができます。またヘッダをクリックして、値を並び替えることもできます。

- ユーザページと同様、このページにも、検索エリアに入力されたテキストに対応する行だけをリストに表示させ、行数を減らすことができる、検索 / フィルタエリアがあります。検索 / フィルタはセッションおよびプロセス名列に対して実行されます。

- ウィンドウに表示されるプロセスをタイプでフィルタするためのボタンが3つあります：

- ユーザプロセス：ユーザセッションにより、またユーザセッションのために作成されたプロセス。このプロセスには以下のアイコンが表示されます：
- 4D プロセス：4D Server エンジンが生成したプロセス。このプロセスには以下のような歯車のアイコンが表示されます：

- 予備プロセス：使用されていないが一時的に保持され、いつでも再利用が可能なプロセス。このメカニズムは 4D Server の反応性を向上させます。このプロセスには以下のようなアイコンが表示されます：
- グループ毎にプロセスを表示 オプションを使用して、4D Server の内部プロセスやクライアントプロセスをグループ化できます。このオプションをチェックすると：
 - 4Dクライアントのプロセス (メインの4Dクライアントプロセスや4Dクライアントの基本プロセス、[ページ 89](#)、"[プロセスタイプ](#)" の段落を参照) は 1 つにグループ化されます。
 - "タスクマネージャ" グループが作成され、タスクを分割するための内部プロセス (共有バランサ、ネットセッションマネージャ、*Exclusive pool worker*) がグループ化されます。
 - "クライアントマネージャ" グループが作成され、クライアントのさまざまな内部プロセスが含まれます。
- ウィンドウの下段には選択したプロセスの稼働状況がグラフィカルに表示されます：



注： Shift キーを押しながら連続した行を、Ctrl (Windows) / Command (Mac OS) キーを押しながら非連続の行を選択できます。

プロセスの稼働状況は、4D Server がこのプロセスのために使用した時間のパーセンテージです。

以前のバージョンの 4D Server のように、ウィンドウにはプロセスごといかの情報が表示されます：

- プロセスタイプ (後述)
- セッション (4D プロセスの場合空白、ユーザプロセスの場合 4D ユーザ名)
- プロセス名
- プロセス番号 (例えば New process 関数で返される値)

注： プロセス番号はサーバ上で割り当てられる番号です。グローバルプロセスの場合、この番号はクライアントマシンで割り当てられた番号と異なる場合があります。

- プロセスの現在の状況
- 作成されてからのプロセスの実行時間 (秒)

■ 4D Server がこのプロセスに使用した時間のパーセンテージ

プロセスタイプ

プロセスタイプはアイコンで識別できます。アイコンの色や形により、プロセスタイプは以下のようになります：

アイコン	プロセスタイプ (説明)
	アプリケーションサーバ
	SQL サーバ
	DB4D サーバ (データベースエンジン)
	Web サーバ
	SOAP サーバ
	保護された 4D クライアントプロセス (接続した 4D の開発プロセス)
	メイン 4D クライアントプロセス (接続した 4D のメインプロセス。クライアントマシン上で作成されたプロセスに対応するサーバプロセス。)
	4D クライアント基礎プロセス (4D クライアントプロセスと並列なプロセス。メイン 4D クライアントプロセスをコントロールする優先プロセス。)
	予備プロセス (以前または未来の "4D クライアントデータベースプロセス")
	4D クライアントプロセス (接続した 4D 上で実行しているプロセス)
	ストアドプロシージャ (接続した 4D により起動され、サーバ上で実行しているプロセス)
	Web メソッド (4DACTION などにより起動)
	SOAP メソッド (Web サービスにより起動)
	SQL メソッド (SQL クエリにより起動)

注：グループ毎にプロセスを表示 オプションがチェックされていると、それぞれの 4D クライアントメインプロセスと 4D クライアント基本プロセスは一緒にグループ化されて表示されます。

管理ボタン

選択されたプロセスに対して動作する 5 つのコントロールボタンがあります。ユーザプロセスに対してのみ使用できる点に注意してください。



- **アボート**: 選択したプロセスをアボートします。このボタンをクリックすると、警告ダイアログが表示され、操作を続行またはキャンセルできません。

注: 確認ダイアログを表示させずに選択したプロセスをアボートできます。これを行うには、Alt (Windows) または Option (Mac OS) を押しながら、このボタンをクリックします。

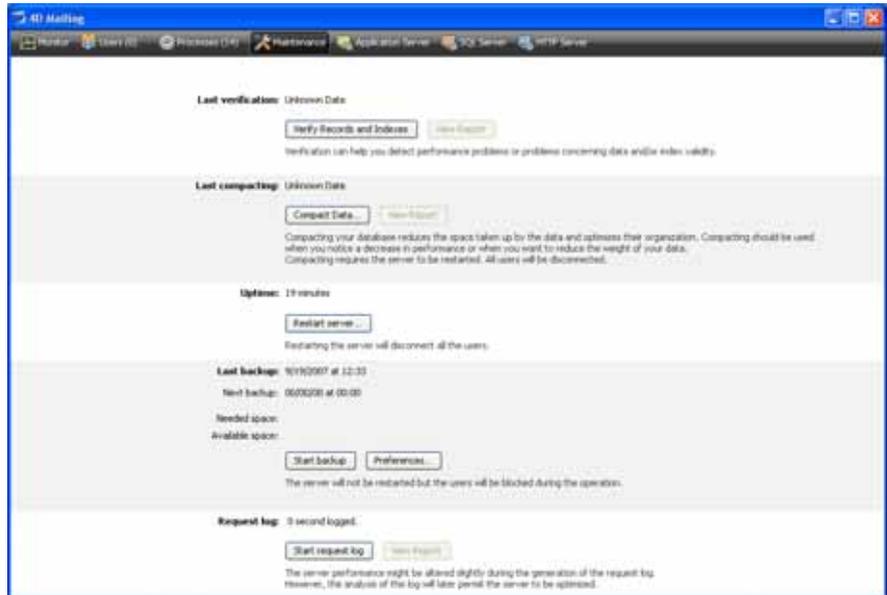
- **一時停止**: 選択したプロセスを一時停止します。
- **再開**: 選択したプロセスの実行を再開します。対象のプロセスは先のボタンを使用するかプログラムにより一時停止状態でなければなりません。そうでなければ、このボタンは効果ありません。
- **デバッグ**: 選択したプロセスのデバッガをサーバマシン上で開きます。このボタンをクリックすると警告ダイアログが表示され、操作を続行またはキャンセルできます。
デバッガウィンドウは、4D コードが実際にサーバマシン上で実行されている場合にのみ、デバッガウィンドウが表示される点に注意してください (例えばトリガのフレームワークや "サーバ上で実行" 属性を持つメソッドの実行時など)。

注: 確認ダイアログを表示させずに選択したプロセスのデバッグを開始できます。これを行うには、Alt (Windows) または Option (Mac OS) を押しながら、このボタンをクリックします。

- **ユーザ表示**: このボタンをクリックすると、選択されたプロセスのユーザを直接、管理ウィンドウのユーザページに表示させることができます。このボタンは最低 1 つのユーザプロセスが選択されている場合に有効になります。

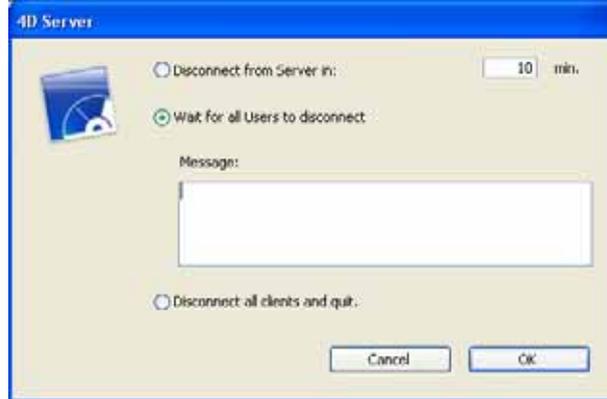
メンテナンス

メンテナンス ページには、データベースの現在の操作に関する情報が表示されます。また基本的なメンテナンス機能にアクセスすることもできます：



- 最後の検証：このエリアには、データベース上で実行された最後の検証の日付、時刻、状況が表示されます。データ検証の方法に関する詳細は、デザインリファレンスを参照してください。
レコードとインデックスを検証 ボタンを使用して、サーバを止めることなく、直接検証を起動できます。検証の間、サーバの動作が遅くなるかもしれないことに留意してください。
データベースのすべてのレコードとすべてのインデックスが検証されます。検証対象を絞り込んだり、追加のオプションを指定したい場合は、Maintenance & Security Center (MSC) を使用します。
検証後、データベースストラクチャファイルと同階層にある Logs フォルダに、XML および HTML ファイル形式でログが作成されます。レポート表示 (クライアントマシンではレポートをダウンロード) ボタンをクリックすると、ブラウザでレポートを参照できます。
- 最後の圧縮：このエリアには、データベースデータに対して実行された最後の圧縮の日付、時刻、そして状況が表示されます。データ圧縮に関する詳細は、デザインリファレンスマニュアルを参照してください。
データ圧縮 ... ボタンを使用して、圧縮処理を起動できます。この操作を行うためには、サーバを停止させる必要があります。ボタンをクリック

すると、4D Server データベースの終了ダイアログが表示され、操作を選択することができます：



データベースが実際に停止された後、4D Server はデータベースデータに対する標準の圧縮操作を行います。追加のオプションを指定したい場合は、Maintenance & Security Center (MSC) を使用します。圧縮が終了すると、4D Server は自動でデータベースを再開し、4D ユーザの再接続が可能になります。

注：圧縮リクエストが 4D クライアントマシンからなされた場合、このマシンは自動で 4D Server により再接続されます。

データベースストラクチャと同階層にある Logs フォルダに、XML および HTML フォーマットでレポートファイルが作成されます。レポート表示 (クライアントマシンでは レポートをダウンロード) ボタンをクリックすると、ブラウザでレポートを参照できます。

- 稼働時間：このエリアには、サーバが開始されてからの稼働時間 (日、時、分) が表示されます。
サーバを再起動 ... ボタンをクリックすると即座にサーバを再起動できます。このボタンをクリックすると、4D Server データベースの終了ダイアログが表示され、操作をどのように中断するか選択できます。
再起動後、4D Server は自動でデータベースを再度開き、4D ユーザの再接続が可能になります。

注：再起動リクエストが 4D クライアントマシンからなされた場合、このマシンは自動で 4D Server により再接続されます。

- 最後のバックアップ：このエリアにはデータベースの最後の自動バックアップの日付と時刻と、環境設定の "スケジュール" で設定された、次回に予定されるバックアップに関する情報が表示されます。

- 次回のバックアップ: 次回の自動バックアップの日付と時刻。
- 必要空き容量: バックアップに必要な計算された空き容量。バックアップファイルの実際のサイズは(圧縮などの)設定や、データファイルの変化により変わります。
- 空き容量: バックアップボリュームの空き容量。

バックアップ開始 を使用して、現在のバックアップパラメタ(バックアップするファイル、アーカイブの場所、オプションなど)を使用したバックアップを開始できます。環境設定 ... ボタンをクリックして、これらのパラメタを確認できます。サーバ上でのバックアップの間、クライアントマシンはブロックされ(ただし接続解除はされません)、新規のクライアント接続はできなくなります。

- リクエストログ: このエリアには、リクエストのログを記録する期間が(それが有効な場合)表示されます。リクエストログファイルには、Webリクエストを除く、サーバが受信したリクエストに関する(時間、プロセス番号、ユーザ、リクエストサイズ、処理時間等、サーバの動作を検証するために使用可能な)情報が格納されます。このファイルは4DRequestsLog_X(Xはファイルのシーケンス番号)という名前で作成され、データベースのLogsフォルダに格納されます。サイズが10MBに達するとファイルは閉じられ、新しいファイルが生成されます。このときシーケンス番号はインクリメントされます。リクエストログ開始 ボタンを使用すると、新しいファイルが作成され、リクエストの記録が有効になります。記録が開始されるとパフォーマンスが著しく低下するため、これはアプリケーションの開発フェーズでのみ使用します。リクエストのログが有効になると、ボタンのタイトルがリクエストログ停止に変わり、いつでもリクエストの記録を停止できます。リクエストのログ記録を再開すると、以前のファイルは消去されることに留意してください。

注: SET DATABASE PARAMETER コマンドを使用してプログラムでログの開始/停止ができます。

レポート表示(クライアントマシンではレポートをダウンロード) ボタンをクリックすると、システムウィンドウが開いて、リクエストログファイルが表示されます。

アプリケーション サーバ

アプリケーションサーバページには 4D Server が公開しているデータベースについての情報がまとめられ、公開を管理できます：



ステータス情報

ページの上部には、4D Server アプリケーションサーバの現在の状況が表示されます。

- 状態：開始または停止状態
- 開始時刻：サーバデータベースの起動日と時刻。これは、データベースが 4D Server によって開かれた日付です。
- 稼働時間：データベースが最後に開かれた時からの経過時間。

新規接続許可 / 拒否ボタン

このボタンは入れ替わり表示され、アプリケーションサーバへの新しいクライアントの接続を管理するために使用できます。

- データベースが公開された時、デフォルトで：
 - ボタンのラベルは " 新規接続を拒否 "
 - ライセンスが許可する限り、新規クライアントは自由に接続が可能
 - データベース名は接続ダイアログに公開される (" 起動時、接続ダイアログにデータベース名を公開 " オプションが環境設定で有効になっている場合)
- 新規接続を拒否ボタンをクリックすると：
 - ボタンラベルが " 新規接続を許可 " に変化

- 新規クライアントは接続不可
- データベース名が接続ダイアログに表示されなくなる
- すでに接続済みのクライアントは接続解除されず、通常通りに操作が可能
- 新規接続を許可 ボタンをクリックすると、データベースはデフォルトに戻ります。

この機能は例えば、サーバ開始直後に管理者が様々なメンテナンス操作（検証や圧縮など）を行うことを可能にします。管理者がクライアント接続を使用する場合、この機能により、一つのクライアントだけがデータを更新できることを確実にできます。

またクライアントマシンがまったく接続されていない状態で行わなければならないメンテナンス操作の準備のために、この機能を使用することができます。

設定

このエリアには、サーバが公開する 4D データベースについての情報（名前、データやストラクチャファイルの場所、データベースログファイルの場所）が表示されます。ストラクチャやデータファイル名をクリックすると、完全なパス名を表示させることができます：

Structure file:	"4D Mailing.4DC" in volume "C:"
Data file:	"4D Mailing.4DD" in volume "C:"
Log file:	<ul style="list-style-type: none"> 📁 C: 📁 Databases 📁 4DMailing 📁 4D Mailing.4dbase 📄 4D Mailing.4DD
Used as service:	
Listening to IP:	
Port:	Copy the path

エリアの下部には、サーバ設定パラメタ（サービスとして起動、ポート、IP アドレス）やクライアント / サーバ接続用の SSL (SQL や Web 接続は除く) の状態が表示されます。

メモリ

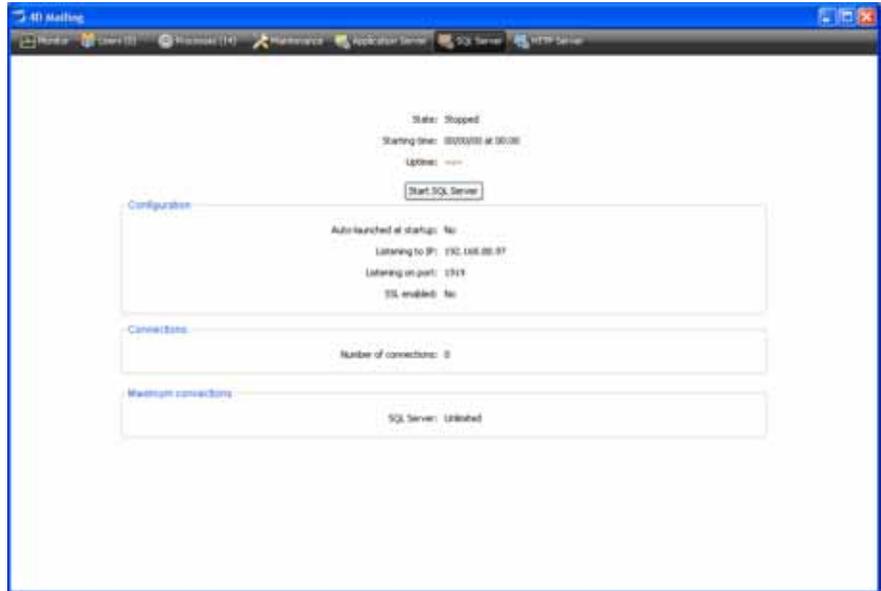
このエリアには総キャッシュメモリ（データベース環境設定で設定されたパラメタ）と使用キャッシュメモリ（必要に応じて 4D Server が動的に割り当て）が表示されます。

最大接続数

許可される同時クライアント接続最大数。この値はサーバマシンにインストールされたライセンスに基づきます。

SQL サーバ

SQL サーバページには、4D Server に統合された SQL についての情報が集められています。また SQL サーバを有効にするためのボタンも含まれています：



状況

ページの上には、4D Server の SQL サーバの現在の状況が表示されます。

- 状態：稼働中または停止中
- 開始時刻：SQL サーバが起動された日付と時刻。この値は、SQL サーバが 4D Server の "On Startup" で起動されていない場合、アプリケーションサーバの起動時刻とは異なることがあります。
- 稼働時間：SQL サーバが開始されてからの経過時間。

SQL サーバ開始 / 停止

このボタンは交互に表示され、4D Server SQL サーバをコントロールするために使用します。

- SQL サーバの状態が "稼働中" の場合、ボタンのタイトルは SQL サーバ 停止になります。このボタンをクリックすると、4D Server SQL サーバは即座に停止します。指定した TCP ポートで受信した外部からの SQL クエリには応答しなくなります。
- SQL サーバの状態が "停止中" の場合、ボタンのタイトルは SQL サーバ 開始になります。このボタンをクリックすると、4D Server SQL サーバは即座に開始します。指定した TCP ポートで受信した外部からの SQL クエ

りに応答します。4D SQL サーバを使用するには、適切なライセンスが必要な点に注意してください。

注：環境設定で設定してアプリケーションの起動時に、またはプログラムを使用して、SQL サーバを自動で開始できます。

設定	<p>このエリアには、SQL サーバ設定のパラメタ (開始時の自動起動、待ち受け IP アドレス、TCP ポート (デフォルトで 19812)、そして SQL 接続用の SSL (4D や Web 接続を除く)) が表示されます。</p> <p>これらの値は 4D の環境設定で変更できます。</p>
接続	4D Server 上で現在開かれている SQL 接続の数。
最大接続数	許可される同時 SQL 接続最大数。この値はサーバマシンにインストールされたライセンスに基づきます。
HTTP サーバ	<p>HTTP サーバページには 4D Server の Web サーバや SOAP サーバに関する情報が集められています。</p> <p>Web サーバは、HTML ページやピクチャなどの Web コンテンツの公開を可能にします。SOAP サーバは Web サービスの公開を管理します。これら 2 つのサーバは、4D Server の内部的な HTTP サーバに依存しています。</p>

このページにはまたこれらのサーバをコントロールするためのボタンが含まれます：



状況

ページの上には 4D Server の HTTP サーバの現在の状態についての情報が表示されます。

- 状態：稼働中または停止中
- 開始時刻：HTTP サーバが起動された日付と時刻。この値は、SQL サーバが 4D Server の "On Startup" で起動されていない場合、アプリケーションサーバの起動時刻とは異なることがあります。
- 稼働時間：HTTP サーバが開始されてからの経過時間。
- 総 HTTP ヒット数：HTTP サーバが開始されてから、サーバが受信したローレベルの HTTP ヒット数。

HTTP サーバ開始 / 停止

このボタンは交互に表示され、4D Server HTTP サーバをコントロールするために使用します。

- HTTP サーバの状態が "稼働中" の場合、ボタンのタイトルは HTTP サーバ停止になります。このボタンをクリックすると、4D Server HTTP サーバは即座に停止します。Web サーバと SOAP サーバはリクエストを受け付けなくなります。
- HTTP サーバの状態が "停止中" の場合、ボタンのタイトルは HTTP サーバ開始になります。このボタンをクリックすると、4D Server HTTP サーバ

バは即座に開始します。Web と SOAP のリクエストが受け入れられます。SOAP サーバは別途停止できることに留意してください ([ページ 99](#)、["SOAP 情報" の段落参照](#))。

- 注:
- HTTP サーバを開始するには適切なライセンスが必要です。
 - 環境設定で設定してアプリケーションの起動時に、またはプログラムを使用して、HTTP サーバを自動で開始できます。

Web 情報

このエリアには 4D Server の Web サーバに関する情報が表示されます。

- Web リクエスト: 許可または拒否。この情報は Web サーバが有効かどうかを示します。Web サーバは直接 HTTP サーバにリンクしているため、Web リクエストは HTTP サーバが開始されていれば受信され、停止されていれば拒否されます。
- 最大接続数: 許可される Web 接続最大数。この値はサーバマシンにインストールされたライセンスに基づきます。

SOAP 情報

このエリアには 4D Server の SOAP サーバに関する情報が表示され、またコントロールボタンが含まれます。

- SOAP リクエスト: 許可または拒否。この情報は SOAP サーバが有効かどうかを示します。SOAP リクエストを受け入れるためには、HTTP サーバが開始され、かつ SOAP サーバが明示的にリクエストを受け入れなければなりません (許可 / 拒否ボタンの説明参照)。
- 最大接続数: 許可される SOAP 接続最大数。この値はサーバマシンにインストールされたライセンスに基づきます。
- SOAP リクエスト許可 / 拒否ボタン: このボタンは交互に表示され、4D Server の SOAP サーバをコントロールするために使用します。このボタンをクリックすると、環境設定の "Web サービス / SOAP" ページの SOAP リクエストを許可設定が変更されます。また環境設定が変更されれば、このボタンのラベルも変わります。
HTTP サーバ停止中に SOAP リクエスト許可ボタンをクリックすると、4D は自動で HTTP サーバを開始します。

HTTP サーバ設定

このエリアには HTTP サーバの設定パラメタや動作に関する情報が表示されます。

- 起動時に HTTP サーバを自動で開始: 4D Server の環境設定で設定されたパラメタ。
- HTTP サーバプロセス (使用済み / 総計): サーバ上で作成された HTTP プロセス数 (現在のプロセス数 / 作成されたすべてのプロセスの総数)。

- キャッシュメモリ (使用済み / 総計): HTTP サーバキャッシュメモリサイズ (キャッシュが実際使用しているサイズ / 環境設定で理論的にキャッシュに割り当てられた最大サイズ)。
- 待ち受け IP、TCP ポート (デフォルトは 80)、(4D と SQL 接続を除く) HTTP 接続用の SSL 有効、そして HTTPS ポート。環境設定の Web/ 設定ページで設定された、HTTP サーバの現在の設定。
- ログファイル情報 : 場所、フォーマット、そして HTTP サーバの次回の自動ログバックアップの日付 (logweb.txt ファイル)。

新しいリモート機能

4D v11 SQL では、4D Server マシンを管理やデータベースのコンパイルを、リモートの 4D (クライアントマシン) から行うことが可能になりました。

クライアントマシンによる 4D Server の管理

リモートの 4D から 4D Server マシンを管理するためには、クライアントマシン上で 4D Server 管理ウィンドウ (ページ 80、" [新しいサーバ管理ウィンドウ](#)" の段落参照) を開きます

クライアントマシンで管理ウィンドウを開く

クライアントマシンでサーバ管理ウィンドウを開くには、Designer または Administrator としてリモートデータベースに接続しなければなりません。そうでないばあい、管理ウィンドウを開こうとするとアクセス権エラー (-9991) が生成されます。

以下の方法を使用できます :

- Help メニューから管理ウィンドウコマンドを選択するか、4D ツールバーの対応するボタンをクリックする :



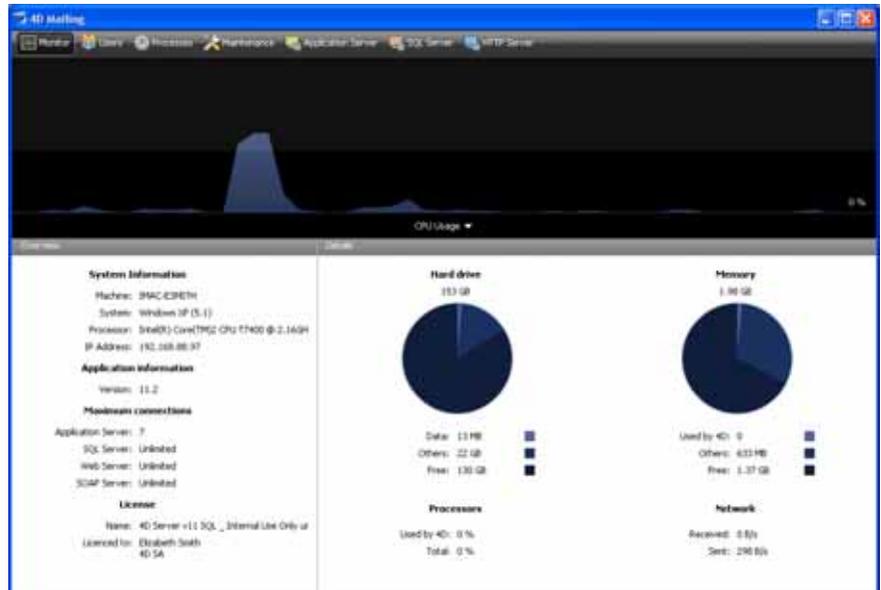
4D ヘルプメニューの新しいコマンド



4D ツールバーの新しいコマンド

- 新しい OPEN ADMINISTRATION WINDOW コマンドを実行する (ページ 42 " [OPEN ADMINISTRATION WINDOW](#)" の段落参照)。

サーバ管理ウィンドウがクライアントマシン上で表示されます：



クライアントマシンでの管理における注意点

サーバ管理ウィンドウを表示しているクライアントマシンは、利用可能なすべての情報にアクセスでき、プロセスを操作したり、サーバを開始/停止したりできます。

しかしながら、クライアントマシン上でサーバ管理ウィンドウが表示されているとき、制限や特定の操作に関する機能があります：

- **プロセス** ページで、ユーザプロセスをデバッグすることはできません (デバッグウィンドウはサーバ上に表示されます)。
- **メンテナンス** ページで、バックアップの環境設定にアクセスできますが、スケジュール、バックアップ、そして復元のページのみを更新できます。
- **メンテナンス** ページで、メンテナンス操作後、レポート表示ボタンはレポートダウンロードボタンとして表示されます。これらのファイルは表示される前に、クライアントマシンのデータベースフォルダにダウンロードされます。

リモートマシンからのコンパイル

リモート接続を使用して、4D アプリケーションをコンパイルできるようになりました。言い換えれば、4D クライアントマシンからコンパイルを行えます。以前のバージョンでは、コンパイルは 4D Developer シングルユーザアプリケーションでのみ可能でした。

注： 他方、カスタム 4D アプリケーションをリモート接続からビルドすることはできません (シングルユーザおよびクライアント / サーバ)。アプリケーションビルダにこの環境からアクセスすることはできません (メニューコマンドが選択不可になります)。

クライアント側で、クライアントインターフェースと操作方法は、シングルユーザバージョンのそれと同じです。コンパイルはデザインメニュー、ツールバー、あるいはコンパイルウィンドウから実行できます：



注： クライアントマシンがコンパイル機能にアクセスするためには、"4D Team Server" ライセンスが必要です。

一度に 1 つのクライアントマシンだけがコンパイルを行えます。クライアントがコンパイルを行っている間、この機能は他のマシンに対してロックされます。同時に他のマシンがデータベースのコンパイルを行おうとすると、警告ダイアログが表示されます。

1 つのクライアントマシンがコンパイルを行っている間、他のクライアントマシンはメソッドやその他のストラクチャ要素を更新できます。このような場合、コンパイルされたコードとインタプリタコードは異なることになり、再度コンパイルが必要になります。

コンパイルされたコードは、コンパイルが実行されるごとにサーバ上の .4DB ファイルに送信されます。

クライアント側で、コンパイルが終了したあと、実行メニューの対応するコマンドを使用して、インタプリタモードまたはコンパイル済みモードでデータベースを再起動できます。コンパイル済み / インタプリタでサーバの再起動がクライアントからリクエストされると、標準のサーバ終了ダイアログが表示され、待ち時間や他のクライアントに送信するメッセージを設定できます。サーバが再起動されると、再起動のリクエスト元のクライアントは自動で再接続されます。

サーバ側で、コンパイル済み / インタプリタでの再起動は標準の開くダイアログボックスを使用する必要があります (開くボタンに関連付けられたポップアップメニュー)。

注：パフォーマンスの理由から、WAN ネットワーク上でのコンパイルは推奨されません (特にデータベースに数多くのメソッドが含まれている場合)。これを行うと、ネットワーク上での数多くのデータ交換が発生します。

リモートマシンからバックアップ環境設定にアクセスする

リモートマシンからほとんどのバックアップ環境設定を更新できるようになりました。スケジュール、バックアップ、そして復元ページのすべてのパラメータを、アクセス権を持つリモートマシンから変更できます。

バックアップされるファイルやディスク上の場所 (設定ページ) は、サーバマシンでのみ変更できます。

4D Server v11 SQL とランゲージ

新しいメソッドプロパティ ([ページ 14 "新しいサーバで実行メソッド属性" の段落参照](#)) に加え、4D Server v11 SQL にはランゲージコマンドの使用に関する最適化が行われました。

サーバ上でフォーミュラによるコマンドの実行

4D v11 で作成されたデータベースでは、最適化の目的で、フォーミュラによるクエリや並び替えのコマンドはサーバ上で実行されるようになりました。結果だけがクライアントマシンに返されます。以下のコマンドが対象です：

QUERY BY FORMULA

QUERY SELECTION BY FORMULA

ORDER BY FORMULA

フォーミュラ中で直接変数を使用されている場合、クライアントマシン上の変数の値を使用してクエリが計算されます。例えば、QUERY BY FORMULA([aTable];[aTable]aField=theVariable) はサーバ上で実行されますが、theVariable 変数の内容はクライアントのものです。

他方この原則は、メソッド内部で変数を使用するようなメソッドを使用したフォーミュラには適用されません。メソッド中で使用される変数の内容はサーバ上のものが使用されます。このコンテキストにおいては、新しいサーバ上で実行のオプションの利用が有効かもしれません。この場合、メソッドをサーバ上で実行させることができ、同時に引数がサーバに送信されます ([ページ 14 "新しいサーバで実行メソッド属性" の段落参照](#))。

4D v11 SQL に変換されたデータベースでは、この新しい機能が既存のアルゴリズムに影響を与えるかもしれないため、デフォルトでこのコンテキストでは、これらのコマンドは引き続きクライアントマシン上で実行

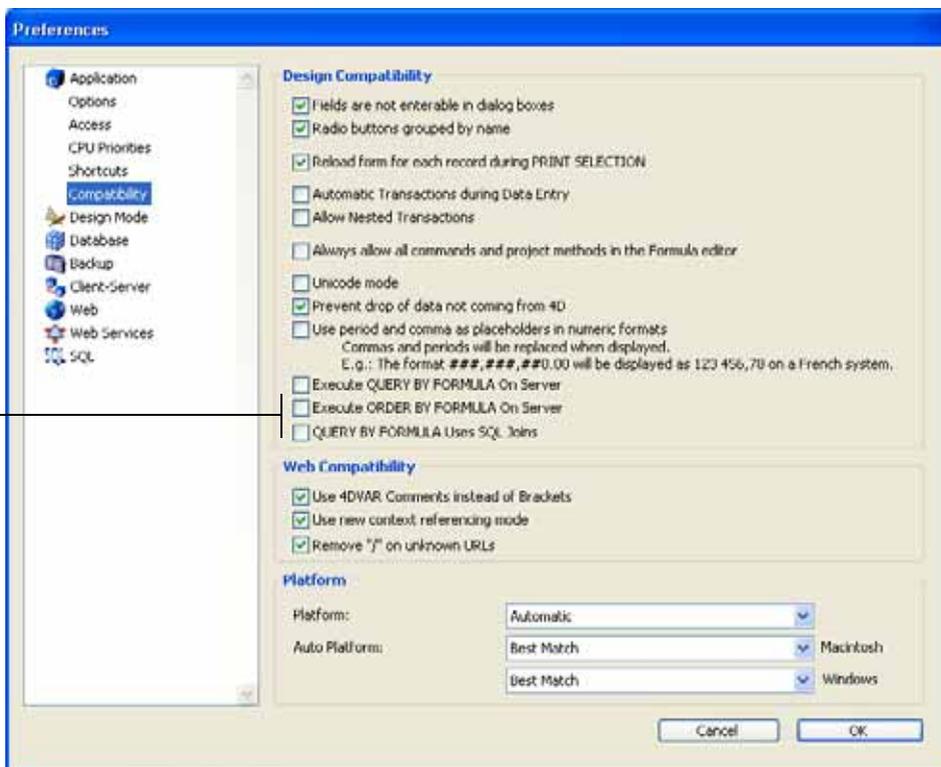
されます。

新しい2つの互換性オプション (環境設定) と新しい2つのセレクトア (SET DATABASE PARAMETER) を使用して、4D v11 に変換されたデータベースの動作を変更できます。

環境設定 (変換されたデータベース)

"互換性" ページの新しい2つのオプションを使用して、変換されたデータベースにおけるフォーミュラを使用したコマンドの動作をグローバルに変更できます：

新しいオプション



- QUERY BY FORMULA をサーバ上で実行：このオプションにチェックされていると、QUERY BY FORMULA と QUERY SELECTION BY FORMULA コマンドはサーバ上で実行されます。
- ORDER BY FORMULA をサーバ上で実行：このオプションにチェックされていると、ORDER BY FORMULA コマンドはサーバ上で実行されます。

新しいセレクトア

新しい2つのセレクトア (定数) が SET DATABASE PARAMETER と Get database parameter コマンドで利用可能になりました。これらはテーブルごとまたプロセスごとに、フォーミュラを使用したコマンドの動作を

指定できます。詳細はページ 46 "SET DATABASE PARAMETER, Get database parameter" の段落を参照してください。

命名セレクションとセット

4D v11 SQL では、セットと命名セレクションを管理するタスクが、透過的にサーバとクライアントで共有されるようになりました。

さらに、名前が \$ で始まるローカル命名セレクションを作成できます。ローカルセットのように、ローカル命名セレクションはそれを作成したマシンおよびプロセスからのみアクセス可能です。

以下の表は、命名セレクションとセットへのアクセスに関する原則を、それらが作成された場所ごとに示します。(両タイプで同じ表です)：

命名セレクションとセットへのアクセス					
	クライアントプロセス	他のクライアントプロセス	他のクライアント	サーバプロセス	他のサーバプロセス
クライアントプロセスで作成					
\$test					
test				(トリガ)	
<>test					
サーバプロセスで作成					
\$test					
test					
<>test					

= アクセス可能

サーバ設定

TCP ポート番号

新しいバージョンの 4D Server は、内部サーバとクライアント間の通信で 3 つの TCP ポートを使用するようになりました：

- SQL サーバ：19812 (デフォルト値、環境設定の "SQL/ 設定" ページで変更可能)。
- アプリケーションサーバ：19813 (デフォルト値、環境設定の "クライアント - サーバ / 設定" ページで変更可能)。
- DB4D サーバ (データベースサーバ)：19814 (デフォルト)。このポートを直接変更することはできません。ただし常にアプリケーションサーバのポート番号に 1 を加えた数となります。

4D クライアントが 4D Server に接続すると、アプリケーションサーバの TCP ポートと通信します (19813、または接続ダイアログで IP アドレスの後の "(コロン)" に続く番号)。指定されたポートを使用した他のサーバへの接続は自動で行われます。個別に指定する必要はありません。

ルータやファイアウォールを経由する接続の場合、これら 3 つのポートは明示的に開かれていなければなりません。

Maintenance & Security Center へのアクセス

リモート 4D/4D Server アーキテクチャのフレームワークで、データベースのメンテナンスは Maintenance & Security Center (MSC) またはランゲージの検証コマンド ("4D 環境" テーマ) を使用して行うことができます。

以下の表では、使用する 4D アプリケーションごとに、これらのツールにアクセスする異なる方法を紹介합니다：

	4D Server	4D リモート モード	4D ローカル モード	4D Volume Desktop	注
Maintenance & Security Center にアクセスする					
ヘルプメニュー /Maintenance Security Center	Yes	-	Yes	-	
"MSC" 標準アクション	-	-	Yes	Yes	(1)
OPEN SECURITY CENTER	-	-	Yes	Yes	
"4D 環境" テーマのランゲージ					
VERIFY DATA FILE	Yes	Yes	Yes	Yes	(2)(3)
VERIFY CURRENT DATA FILE	Yes	-	Yes	Yes	(3)
データファイルの圧縮	Yes	Yes	Yes	Yes	(2)(3)

(1) カスタムメニューとフォームをサーバ上で使用することはできません。

(2) カレントデータファイルには適用されません。

(3) 'サーバ上で実行' により、サーバで実行が可能。

5

プラグイン

4D Internet Commands と 4D View v11.2 プラグインに新しい機能が追加されました。

4D Internet Commands

4D Internet Commands は UTF-8 でエンコードしたメールメッセージの送信をサポートします。これを行うために、SMTP_SetPrefs コマンドに新しいセレクタが追加されました。

SMTP_SetPrefs

SMTP_SetPrefs (lineFeed; bodyType; lineLength) → 整数

SMTP_SetPrefs コマンドで、送信するメッセージに UTF-8 文字セットを利用することができるようになりました。文字セット (Body-Content-Type) とエンコードタイプ (Content-Transfer-Encoding) に関連する 2 つの新しいセットを bodyType 引数に渡すことができます：

15: UTF-8 & quoted-printable

16: UTF-8 & base64

SMTP_GetPrefs コマンドにもこの変更が適用されます。

- 以下のコードは quoted-printable でエンコードした UTF-8 メッセージを送信します：

```
$err:=SMTP_SetPrefs(-1;15;-1)
```

```
$err:=SMTP_QuickSend(...)
```

テーマ : IC Send Mail.

4D View

4D View エリアのペインを固定することができるようになりました。この新しい機能にはメニューコマンドおよびプログラムからアクセスできます。

ペインの固定と固定解除

4D View の新しいバージョンでは、4D View エリアのペインを固定できるようになりました。ペインが固定されると、ユーザがどのようにスプレッドシートをブラウズするかにかかわらず、そのペインは常にスクリーンに表示されます。

固定されると、ペインのサイズを変更したり、ユーザが手動で削除したりできなくなります。他の 4D View エリアからスクロールを行ってこのコンテンツにアクセスすることもできなくなります。例えば、エリアの先頭 3 行を固定ペインにすると、4D View エリアの他のペインからは 4 行目より上に行くことはできません。

固定ペインはデフォルトではロックされていません。内容やフォーマット等を入力したり変更したりできます。

ひとつの 4D View エリアは最大 2 つの固定ペイン、最上部の水平ペインおよびエリア左の垂直ペインを持つことができます。

固定ペインと標準ペインとは、細線のスプリッタにより、視覚的に異なるものとなります：

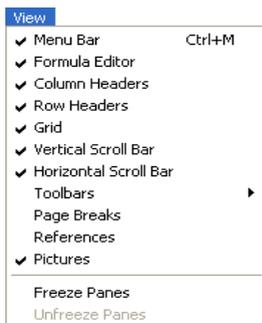
固定ペインスプリッタ

	A	B	C	D	E	F	G
1		Theater 1	Theater 2	Theater 3	Theater 4	Theater 5	
4	February	15	45	30	20	5	
5	March	10	20	15	30	15	
6	April	25	15	10	15	5	
7	May	30	30	15	25	25	
8	June	15	15	20	25	30	
9	July	45	15	25	5	15	
10	August	40	5	15	25	25	
11	September	10	15	20	25	30	
12	October	50	15	15	25	5	
13	November	15	25	15	60	5	
14	December	45	35	25	15	15	

同じドキュメントに標準ペインと固定ペインを混在させることはできません。4D View エリアは "標準ペイン" モードあるいは "固定ペイン" モードで操作します。"固定ペイン" モードで動作するとき、ペインを追

加することはできません。新しいスプリッタの追加を可能にするには、ペインの固定を解除しなければなりません。

新しいメニューコマンド 2つの新しいコマンド、ペインを固定とペインの固定解除が表示メニューに追加され、固定ペインを管理できます：



これらのコマンドは交互に、そして 4D View エリアのコンテキストにより有効になります。

Note: 2つの新しいコマンドを使用して、固定ペインを管理することもできます (以下の段落参照)。

- ペインを固定 コマンドは以下の場合に有効になります：
 - 4D Viewエリアが1つの水平または/および垂直スプリッタをもつとき。
 - ペインが固定されていないとき。

このコマンドを選択すると、水平および垂直ペインが現在の場所で固定されます。このエリアにペインを追加したり削除したりすることはできなくなります。

- ペインの固定解除 コマンドは、4D View エリア内のペインがすでに固定されているときに有効になります。このコマンドを選択すると、エリアのペインは標準状態に戻り、それらを追加したり削除したりできるようになります。

新しいコマンド

いくつかの新しい 4D View ランゲージコマンドを使用して、プログラムでペインをフリーズしたり解除したりできます。既存のコマンドも、この新しい機能をサポートできるよう変更されました。

PV FREEZE PANES

PV FREEZE PANES(area; mode)

引数	型	説明
----	---	----

area	倍長整数	→	4D View エリア
mode	倍長整数	→	0= スクロールをロック, 1= スクロールとヘッダの更新をロック

PV FREEZE PANES コマンドは、4D View エリアの 1 番目の水平ペインおよび / または 1 番目の垂直ペインを固定します。コマンドが機能するためには、最大 1 つの垂直スプリッタおよび / または 1 つの水平スプリッタがエリアにあり、かつ固定されていない状態でなければなりません。

mode 引数を使用して、エリアに適用するロックのタイプを指定できます :

- mode = 0: ペインの内側のスクロールのみが固定されます。このロックは表示メニューのペインを固定コマンドで実行されるものです。
- mode = 1: ロックが拡張され、スクロールに加え、ヘッダの更新 (スタイル、フォントサイズ等) にも影響します。またペインのセルをクリックすると、すべての列と行が選択されます。

エリアにペインスプリッタが含まれていないか、1 つ以上の垂直もしくは水平ペインスプリッタがある場合、コマンドは何も行わず、OK システム変数が 0 に設定されます。

コマンドが正しく実行されると、OK 変数が 1 に設定され、表示メニューが更新されます。つまりペインを固定コマンドが選択不可となり、ペインの固定解除コマンドがアクティブになります。

エリアを " 標準ペイン " モードに戻すには、PV UNFREEZE PANES コマンドを実行するか、ペインの固定解除メニューコマンドを選択します。

- 以下のコードは、エリアに水平ペインが 2 つあるかを確認し (1 スプリッタ = 2 ペイン)、拡張モードで固定します :

```
$nbpanes:=PV Get area property(area;pv hor pane count)
If ($nbpanes=2)
  PV FREEZE PANES(area;1)
End if
```

参照 : [PV UNFREEZE PANES](#)

テーマ : PV ペイン .

PV UNFREEZE PANES PV UNFREEZE PANES (area)

引数	型	説明
area	倍長整数	→ 4D View エリア

PV UNFREEZE PANES コマンドは、4D View エリアを "標準ペイン" モードに変更します。このモードではペインの追加や削除、スプリッタの移動などが可能です。

このコマンドが動作するためには、エリアのペインが、[PV FREEZE PANES](#) コマンド、または表示メニューのペインを固定コマンドを使用して事前に固定されていなければなりません。

エリアにスプリッタが含まれていないか、固定されていない場合、コマンドは何も行わず、OK 変数に 0 が設定されます。

コマンドが正しく実行されると、OK 変数に 1 が設定され、表示メニューが更新されます。つまりペインの固定解除コマンドが選択不可となり、ペインを固定コマンドがアクティブになります。

- 以下の例は、すでに固定されていれば、エリアのペイン固定を解除します：

```
PV GET COMMAND STATUS(area;pv cmd unfreeze panes;$status;
$check;$name)
If ($status=1)
  PV UNFREEZE PANES(area)
End if
```

参照：[PV FREEZE PANES](#)

テーマ：PV ペイン

ランゲージの変更

固定ペイン機能をサポートするために、さまざまな変更が 4D View ランゲージに対し行われました。

新しいエラーコード

PV ADD HOR SPLITTER や PV ADD VERT SPLITTER コマンドが固定ペインモードのエリアで実行されると、エラー 92 が生成されます：

```
92No splitter can be added when panes are frozen
```

新しい定数

以下の定数が "PV Commands" テーマに追加されました：

定数	型	値
pv cmd freeze panes	倍長整数	171
pv cmd unfreeze panes	倍長整数	172

ペイン削除コマンド

PV REMOVE HOR SPLITTER(area;1) と PV REMOVE VERT SPLITTER(area;1) 文を固定エリアのコンテキストで実行すると、固定ペインが取り除かれ、"標準ペイン" モードに戻ります。

ペインプロパティ管理コマンド "固定ペイン" モードのエリアのコンテキストでは、ペインのプロパティを管理するコマンド (PV SET HOR PANE PROPERTY、PV SET VERT PANE PROPERTY、PV Get hor pane property、そして PV Get vert pane property) は以下のプロパティのみを使用できます：

- pv pane columns count
- pv pane rows count
- pv pane first column
- pv pane first row

他のプロパティは無効となります。