











4D v15 R3 - アップグレード

-  デザインモード
-  ランゲージ
-  Web サーバー
-  4D Write Pro
-  最適化
-  4D Developer Edition 64-bit版(プレビュー版)

デザインモード

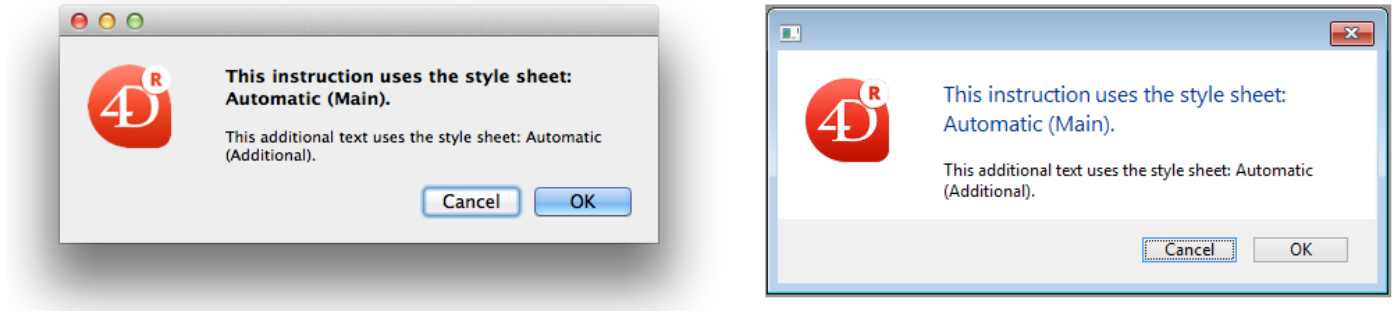
-  新しい自動スタイルシート
-  リストボックス：シングルクリック編集モード
-  タイムスタンプ付きのメンテナンスログファイル名
-  重複した値の管理の改善

新しい自動スタイルシート

概要

4D v15 R3では、二つの"自動"スタイルシートを扱う事ができます。これにより、カレントプラットフォームのガイドラインに沿った、一般的なダイアログボックスをデザインする事ができます。

これらの新しいスタイルシートはそれぞれ**自動(メイン)**と**自動(追加)**と呼ばれ、前者はインターフェースウィンドウ内のメインのテキストに使用され、後者はその追加の情報に使用されます。以下はこれらの新しいスタイルシートを使ったダイアログボックスの一例です:



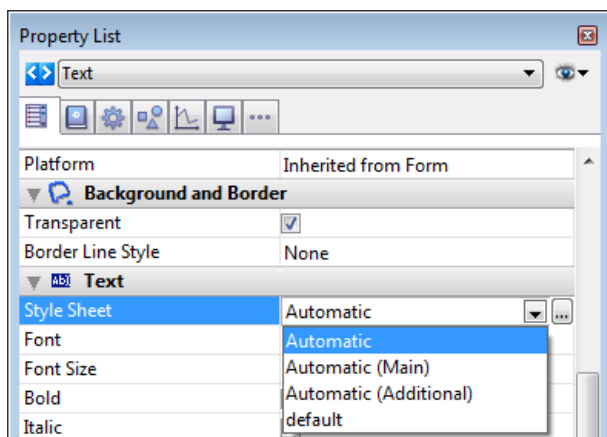
新しい自動スタイルシートを選択する

新しい自動スタイルシートは以下のオブジェクトにおいてサポートされています:

- スタティックテキスト
- フィールド
- 変数

これらのオブジェクトにおいてはフォームエディターのプロパティリスト、またはランゲージを使用する事によって利用可能です(使用するコマンドについての詳細な情報については、[新しいFont Style定数の章](#)を参照して下さい)。

フォームエディター内に置いては、**自動(メイン)**と**自動(追加)**のオプションはプロパティリストの標準の**スタイルシート**ポップアップメニュー内にあります:



新しい自動スタイルシートのプロパティ

新しい自動スタイルシートはオブジェクトテキストに対して以下のプロパティを設定します:

- フォント
- フォントサイズ
- フォントカラー

これらのプロパティはプラットフォームとシステム言語に応じて、デザインモードまたはアプリケーションモードにおいて動

的に定義されます。

既存の"自動"スタイルシートについては、フォームエディター内においてスタイルシートプロパティの一つを変更してしまうと、それ以降は動的には作動しません。ただし、スタイルのプロパティ(**太字**、**イタリック**、**下線付き**)についてはその自動機能を損なう事無く、カスタムのもを適用する事が出来ます。

📄 リストボックス : シングルクリック編集モード

新しいオプション**シングルクリック編集**を使用すると、リストボックスの簡単な編集モードを有効化する事ができます。このオプションはリストボックスオブジェクトのプロパティリストの"入力"エリアにあります:

▼ 📄 Entry	
Focusable	<input checked="" type="checkbox"/>
Hide focus rectangle	<input type="checkbox"/>
Tabable	<input checked="" type="checkbox"/>
Single Click Edit	<input checked="" type="checkbox"/>

このオプションがチェックされていると、リストボックスのこのエリアがその直前に選択されていたかどうかに関わらず、シングルクリックだけでリストボックスセルを編集モードへとスイッチします。またこのオプションが有効化されていた場合、セレクションモードが"なし"に設定されていた場合でもセルが編集可能になるという点に注意して下さい。

このオプションがチェックされていない場合、ユーザーがセルの中身を編集するためには、まずセルの行を選択し、そのあとセルをクリックする必要があります。これは以前のバージョンにおける標準の仕様となります。互換性のために、このオプションはデフォルトではチェックが外れています。

📄 タイムスタンプ付きのメンテナンスログファイル名

概要

4D v15 R3以降、MSCまたは4D Server管理ウィンドウを通して行われたメンテナンスオペレーション中に生成されたログファイルについては、名前にタイムスタンプがつくようになりました。例えば、"Verify"というオペレーション中に生成されたログファイルには、以下の様な名前がつけられます：

```
Invoices_Verify_log_2015-10-23 17-50-13.xml
```

このうち、"2015-10-23 17-50-13" がログファイルのタイムスタンプです。

以前のバージョンでは、ログファイルはタイムスタンプが押されませんでした。毎回メンテナンスオペレーションが始まるたび、以前のログファイルは(あれば)、新しいもので上書きされていました。たとえそれがユーザーキャンセルであったとしてもです。この新機能のおかげで、ログファイル名は全て固有の名前になりました。同じオペレーションに対して、データベースの"Logs"フォルダ内に複数のログファイルを保存する事が可能になりました。また、タイムスタンプ値のおかげで、ログファイルの特定もしやすくなりました。

詳細

ログファイルは、MSCウィンドウまたは4D Server 管理ウィンドウ(メンテナンスページ)から生成されたときには自動的にタイムスタンプが押されます。ランゲージを使用して生成されたログファイルにもタイムスタンプを押すためには、この機能を有効化するオプションを渡す必要があります(以下参照)

これに関連するのは以下のオペレーションです：

- "検証",
- "修復" (4Dのみ)
- "圧縮"

ファイル名は、以下のパターンに基づいて生成されます：<StructureName>_<Operation>_Log_<yyyy-mm-dd hh-mm-ss>.xml

- 以前のバージョンでは以下ようになっていました：
 - <StructureName>は拡張子を除いた、4Dストラクチャーファイルの名前でした。例："Invoices"
 - <Operation> には"検証"、"修復"または"圧縮"が入ります。
- 新しい<yyyy-mm-dd hh-mm-ss>の部分はタイムスタンプであり、メンテナンスオペレーションが開始したローカルのシステム時刻に基づきます。

ユーザーが、MSCウィンドウの**ログファイルを開く**ボタン、あるいは4D Serverメンテナンスページの**レポートを見る**ボタンをクリックすると、一番直近のファイルが開かれます。

ログファイル名が全てユニークになったことにより、"Logs"フォルダの中身は自動的に整理・削除されないという点に注意して下さい。4Dにおいても4D Serverにおいても、データベース管理者が責任を持って、古いログファイルを必要に応じて削除するようにして下さい。

ランゲージコマンド

メンテナンスのランゲージコマンド(**Compact data file**、**VERIFY DATA FILE**そして**VERIFY CURRENT DATA FILE**)もこの新機能をサポートしますが、そのためには新しい任意の定数**Timestamp log file name**で有効化される必要があります。

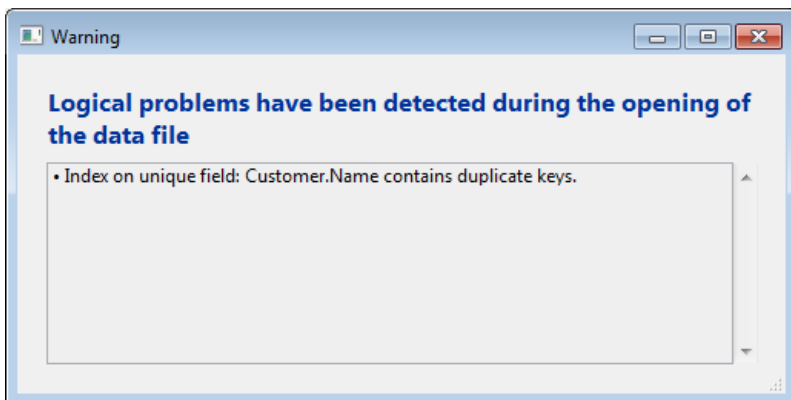
詳細な情報については、アップグレードリファレンスの"**データファイルメンテナンスの新しい定数**"の章を参照して下さい。

重複した値の管理の改善








4D v15 R3以降、"重複不可"に設定されたフィールドで見つかった重複する値に関する情報が改善されました(このような重複した値は、例えばストラクチャーレベルでの変更、あるいは異なるデータファイルを使用することにより発生する可能性があります)。

新しい機能により、4Dは例えば以下の様な状況を手助けする情報をデベロッパに提示する事ができます:

- 重複不可フィールドにおいて値の重複が検知された際により多くの情報を提供します:
 - MSCまたは**VERIFY DATA FILE**のようなコマンドを使用した際には、それによって生成されたログには違反しているテーブルとフィールド、そして重複している値も表示します。
 - データ入力時、"重複キー"エラーダイアログボックスにも違反しているテーブルとフィールド、そして重複している値が表示されます。
 - **GET LAST ERROR STACK**コマンドは、どのような重複についてもその詳細な情報を含むようになりました。
- 4Dがデータファイルを開く際、インデックスを生成(あるいは再生成)する必要があるとき、関連したフィールドのうち重複不可として宣言されているものの中では重複した値は自動的に検知されます。この場合、特別な警告ダイアログボックスがデータベースを開く前に表示され、重複した値を特定・削除するために必要な情報をユーザーに提供します:



ランゲージ

-  QR GET INFO COLUMN
-  QUERY BY ATTRIBUTE
-  データファイルメンテナンスの新しい定数
-  新しいFont Style定数
-  CONVERT COORDINATES
-  QR MOVE COLUMN
-  QR NEW AREA

QR GET INFO COLUMN

QR GET INFO COLUMN (area ; colNum ; title ; object ; hide ; size ; repeatedValue ; displayFormat {; resultVar})

引数	型	詳細
area	Longint	-> エリアの参照
colNum	Longint	-> カラム番号
title	String	<- カラムタイトル
object	Text	<- フィールド名またはカラムに割り当てられたフォーミュラ
hide	Longint	<- 0 = 表示、1 = 非表示
size	Longint	<- カラムサイズ
repeatedValue	Longint	<- 0 = 繰り返しなし、1 = 繰り返しあり
displayFormat	Text	<- データの表示フォーマット
resultVar	テキスト	<- フォーミュラ変数の名前

説明

QR GET INFO COLUMN コマンドは新しい任意の引数、*resultVar*を返すようになりました。

これが渡された場合、この変数は、フォーミュラカラムがあった場合にクイックレポートエディターによってそれに自動的に割り当てられた変数名を返します。最初のフォーミュラカラムは"C1"、二つ目は"C2"、といった具合です。4Dは、レポートを生成する際に、カラムのフォーミュラの最後の実行から得られた結果を保存するのにこの変数を使用します。

例題

以下のようなレポートを作成した場合を考えます：

	[People]LastName	[People]FirstName	C1	[People]City	[People]Salary
Title	LastName	FirstName	Age	City	Salary
Detail					
Grand total					

その場合、以下のようにコードを書く事ができます：

```
C_TEXT ($vTitle; $vObject; $vDisplayFormat; $vResultVar)
C_LONGINT ($area; $vHide; $vSize; $vRepeatedValue)
QR GET INFO
COLUMN ($area; 3; $vTitle; $vObject; $vHide; $vSize; $vRepeatedValue; $vDisplayFormat; $vResultVar)
// $vTitle = "Age"
// $vObject = "[People]Birthdate-Current date"
// $vHide = 0
// $vSize = 57
// $vRepeatedValue = 1
// $vDisplayFormat = ""
// $vResultVar = "C1"
```

新しいクイックレポートについて

4D v15 R2では、将来の4Dリリースに適合した新しいクイックレポートも提供しています。これは4D v15 R2 64-bit版にデフォルトで含まれていますが、32-bit版でテストする事も可能です。詳細な情報に関しては、[新しいクイックレポートエディター\(プレビュー版\)](#)の章を参照して下さい。

QUERY BY ATTRIBUTE

```
QUERY BY ATTRIBUTE ( aTable {; conjOp}; objectField ; attributePath ; queryOp ; value {; *} )
```

説明

このコマンドは、ドキュメントがオブジェクトフィールドに保存されていた場合、4D Write Proカスタム属性をサポートするようになりました。詳細な情報については、[Storing 4D Write Pro documents in 4D Object fields](#)の章を参照して下さい。

例題

以下の例では、レコードを選択するためにObjectフィールドを含んだテーブルをクエリします:

```
QUERY BY ATTRIBUTE ([MyDocuments]; [MyDocuments]My4DWP;"myatt_Category";="Memo")  
//MyDocuments内で、(4D Write Proエリアに関連づけられた) My4DWPオブジェクトフィールドの中で  
// "myatt_Category"カスタム属性が"Memo"の値を含む全てのレコードを選択
```

データファイルメンテナンスの新しい定数

"データファイルメンテナンス"定数テーマに新しい定数が追加されました:

定数	型	値	詳細
Timestamp log file name	倍長整数	262144	ログファイル名の自動タイムスタンプを有効化します

この新しい定数は、以下のメンテナンスコマンドの`options`引数において使用する事ができます:

- **Compact data file**
- **VERIFY DATA FILE**
- **VERIFY CURRENT DATA FILE**

これらのコマンドにこのオプションが渡された場合、ログファイルの名前はタイムスタンプがつくので、既存のログファイルを上書きする事がなくなります。デフォルトではこのオプションは渡されていないので、生成されたログファイルはその前のログファイルを上書きします。

ログファイルが生成されると、選択されたオプションとは関係なく、そのファイルへのパスはコマンド実行後に**Document**システム変数へと返されます。

タイムスタンプについての詳細な情報に関しては、[タイムスタンプ付きのメンテナンスログファイル名](#)の章を参照して下さい。

新しいFont Style定数

4D v15 R3では、4Dデベロッパが一般的なダイアログボックスをネイティブなインターフェースでデザインするのを手助けする新しい自動スタイルシートを用意しています。これについての詳細な情報に関しては、[新しい自動スタイルシート](#)の章を参照して下さい。

これらのスタイルシートは、"フォントスタイル"テーマに追加された二つの新しい定数を使用する事によって、プログラミングで管理する事ができます:

定数	型	値
Automatic style sheet_main	文字列	__automatic_main_text__
Automatic style sheet_additional	文字列	__automatic_additional_text__

この新しい定数は、以下のコマンドで使用する事ができます:

コマンド	引数
GET STYLE SHEET INFO	<i>styleSheetName</i>
OBJECT Get style sheet	<i>Function result</i>
OBJECT SET STYLE SHEET	<i>styleSheetName</i>

注: 互換性上の理由から、これらの新しいスタイルシートは**LIST OF STYLE SHEETS**コマンドでは返されません。ですが使用する事は可能です。

例題

メインの自動スタイルシートをフォームの変数エリアに適用したい場合を考えます:

```
OBJECT SET STYLE SHEET(*;"myTextVar";Automatic style sheet_main)
```

CONVERT COORDINATES (xCoord ; yCoord ; from ; to)

引数	型		説明
xCoord	倍長整数変数	→	Horizontal coordinate of a point (initial)
		←	Horizontal coordinate of a point (converted)
yCoord	倍長整数変数	→	Vertical coordinate of a point (initial)
		←	Vertical coordinate of a point (converted)
from	倍長整数	→	Coordinates system to convert from
to	倍長整数	→	Coordinates system to convert to

説明

テーマ: Windows

新しい**CONVERT COORDINATES** コマンドは、ある点の(x;y)座標を他の座標系へと変換します。サポートされる入力と出力の座標系は、フォーム(およびサブフォーム)、ウィンドウ、そしてスクリーンです。例えばこのコマンドを使用して、サブフォームに属するオブジェクトのメインフォーム内での座標を取得することができます。これによりどのようなカスタムのポジションでもコンテキストメニューを作成するのが容易になります。

xCoordとyCoordには、変換したい点の(x;y)座標を変数として渡します。コマンドが実行されると、これらの変数には変換後の値が格納されます。

from 引数には、入力点で使用している、元の座標系を渡します。そしてto引数にはそれらの座標を変換したい座標系を渡します。どちらの変数も、"Windows"テーマに追加された以下の定数値のうちのどれか一つを使用することができます:

定数	型	値	補足
XY Current form	Longint	1	原点はカレントフォームの左上端
XY Current window	Longint	2	原点はカレントウィンドウの左上端
XY Screen	Longint	3	原点はメインスクリーンの左上端(SCREEN COORDINATES コマンドと同じ)
XY Main window	Longint	4	原点はメインウィンドウの左上端(Windows)/XY Screenと同じ(OS X)

注: "Windows" 定数テーマは、以前の"Window kind"テーマの新しい名前であり、4D v15 R3以降で一般的なものになりました。

このコマンドがサブフォームの、あるいはサブフォーム内のオブジェクトのメソッドから呼び出され、セレクターの一つがXY Current formであった場合、座標系はサブフォーム自身からみた相対座標になります(親フォームではありません)。

もしフォームウィンドウの位置から、あるいは位置を変換する場合(例えば**GET WINDOW RECT**の結果を変換、あるいは**Open form window**に渡した値に変換する様な場合)、XY Main windowを使用する必要があります(Windows上のウィンドウコマンドで使用される座標系であるため)。同じ目的でOS X上で使用する事もできますが、これはXY Screenと同じことになります。

from変数がXY Current formであり、変換したい点がリストフォームのボディ部分にある場合、その結果はコマンドの呼び出しコンテキストによって変化します:

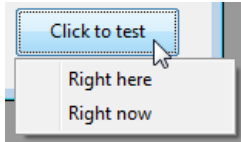
- コマンドがOn Display Detail イベントで呼び出された場合、変換後の点はスクリーン上に描画されたレコードの表示内に位置します。
- コマンドがOn Display Detail イベント外で呼び出され、かつレコードが編集集中であった場合、変換後の点は編集集中のレコードの表示内に位置します。
- それ以外の場合、変換後の点は最初のレコードの表示内に位置します。

例題 1

"MyObject"というオブジェクトの左下端にポップアップメニューを表示したい場合を考えます。

```
// OBJECT GET COORDINATES はカレントフォームの座標系でのみ動作する
// 動的なポップアップメニューはカレントウィンドウの座標系を使用する
// そのためこれらの値を変換する必要がある
```

```
C_LONGINT($left;$top;$right;$bottom)
C_TEXT($menu)
OBJECT GET COORDINATES(*,"MyObject";$left;$top;$right;$bottom)
CONVERT COORDINATES($left;$bottom;XY Current form;XY Current window)
$menu:=Create menu
APPEND MENU ITEM($menu;"Right here")
APPEND MENU ITEM($menu;"Right now")
Dynamic pop up menu($menu;"";$left;$bottom)
RELEASE MENU($menu)
```




例題 2

マウスカーソルの位置にポップアップメニューを開きたい場合を考えます。Windowsの場合、**GET MOUSE**コマンド(* 引数付き)はMDIウィンドウの位置に基づいた値を返すため、座標系を変換する必要があります:

```
C_LONGINT($mouseX;$mouseY;$mouseButtons)
C_LONGINT($window)
GET MOUSE($mouseX;$mouseY;$mouseButtons)
CONVERT COORDINATES($mouseX;$mouseY;XY Current window;XY Main window)
>window:=Open form window("PopupWindowForm";Pop up form window;$mouseX;$mouseY)
DIALOG("PopupWindowForm")
CLOSE WINDOW($window)
```

Web サーバー

 アップロード時のチャンクドトランスファーエンコーディングのサポート


アップロード時のチャンクドトランスファーエンコーディングのサポート


4DにビルトインされているWebサーバーは、どのようなWebクライアントからでも、チャンクドトランスファーエンコーディングでのファイルのアップロードをサポートするようになりました。チャンクドトランスファーエンコーディングとはHTTP/1.1で説明されているデータ転送機構です。これは最終的なデータサイズを知る事なく、データを一連の"チャンク"(部分)として転送することが出来るようにするものです。

注: 4D Webサーバーでは、既にサーバーからWebクライアントへのチャンクドトランスファーエンコーディングはサポートしていました([WEB SEND RAW DATA](#)を参照して下さい)。

チャンクドトランスファーについてのクライアント側の実装についてのより詳細な情報は、[RFC2616](#)、またはそれに関連した[Wikipedia](#)のページを参照して下さい。

4D Write Pro

 Storing 4D Write Pro documents in 4D Object fields

 フォーム内の4D Write Proエリアを印刷する

Storing 4D Write Pro documents in 4D Object fields

Starting with 4D v15 R3, you can store your 4D Write Pro documents automatically in the 4D data file. If you created a 4D Write Pro area on a form and created an Object field to store the area's contents, any text entered in the area is saved automatically with each record when the record is validated. You can also add custom attributes to 4D Write Pro areas and use them for queries.

This section describes the following features:

- Binding a 4D Object field to a 4D Write Pro area in a form
- Setting, getting, and querying custom attributes of stored 4D Write Pro documents using the **OB SET**, **OB Get** standard object commands and **QUERY BY ATTRIBUTE**.

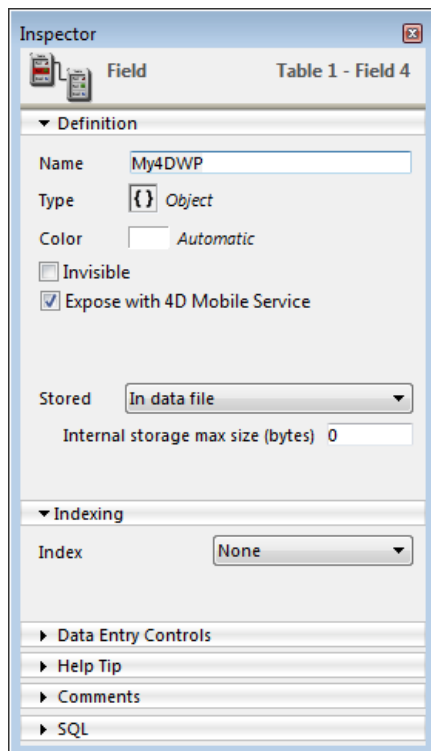
Binding a 4D Object field to a 4D Write Pro area

To bind a 4D Write Pro area with a 4D Object field, you just need to reference the field in the Variable Name property of the area.

Creating the Object field in the Structure

In your database structure, any 4D Object field can be used to store 4D Write Pro documents. As with any Object field, you just have to define its standard properties, according to your needs:

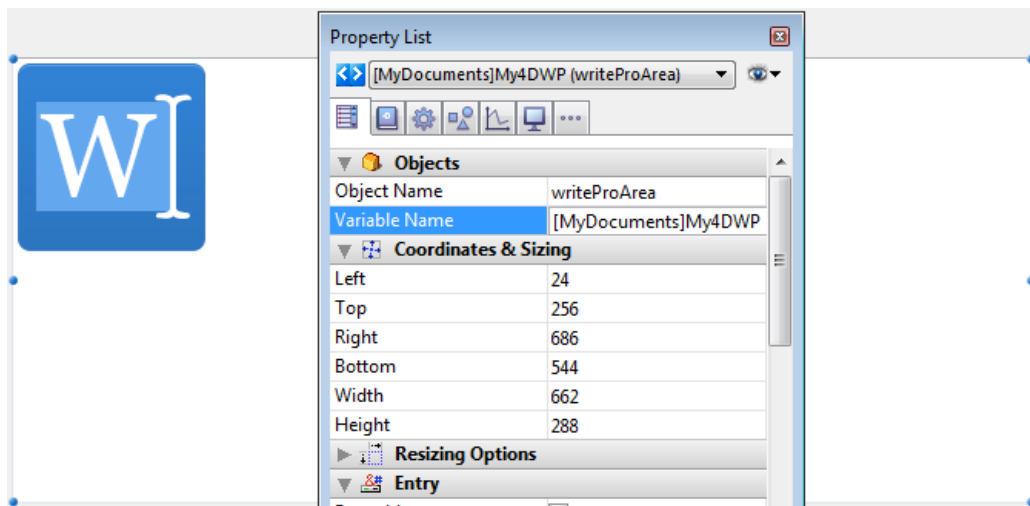
- the field name,
- its attributes, such as "Expose with 4D Mobile Service," as well as its index,
- its storage option (for more on this, see [External data storage](#)).



Assigning the Object field to the 4D Write Pro area

Once you have defined an Object field to store your 4D Write Pro document, you just need to reference it in the form containing the area. You can use any table or a project form.

In the Form editor, enter the field name using the standard "[Table]Field" notation in the **Variable Name** area of the Property list for the 4D Write Pro area:



Your 4D Write Pro area is then bound to the field, ensuring that its contents will be saved automatically with each record. Note that if you do not use the 4D automatic action buttons, you will have to save the area manually using 4D commands.

Using custom attributes

When 4D Write Pro areas are stored in Object fields, you can save and read any custom attributes with the 4D Write Pro document, such as, for example, the writer's name, the document category, or any additional information you may find useful. You can then query your custom attributes to select records matching the criteria.

- Custom attributes are not exported with the **WP EXPORT DOCUMENT** or **WP EXPORT VARIABLE** commands
- On the other hand, custom attributes are exported when converting a 4D Write Pro Object field to JSON using the **JSON Stringify** command (along with the 4D Write Pro "title" attribute).

To set or get custom attributes, you just need to use the standard **OB Get** and **OB SET** commands.

For example, in the form method, you can write:

```
If(Form event=On Validate)
  OB SET([MyDocuments]My4DWP;"myatt_Last edition by";Current user)
  OB SET([MyDocuments]My4DWP;"myatt_Category";"Memo")
End if
```

You can also read custom attributes of the documents:

```
vAttrib:=OB Get([MyDocuments]My4DWP;"myatt_Last edition by")
```

If you have saved custom 4D Write Pro attributes in your data file, you can query these attributes to create a selection of records containing the appropriate attribute value. In the following example, you query the table containing the Object field to select records:

```
QUERY BY ATTRIBUTE([MyDocuments];[MyDocuments]My4DWP;"myatt_Category";="Memo")
//selects all records in MyDocuments whose "myatt_Category" custom attribute has the value
"Memo"
//in the My4DWP Object field (bound to a 4D Write Pro area)
```

Note about custom attribute names: Since custom attributes share the same naming space as 4D Write Pro internal attributes, we strongly recommend that you use prefixes when defining your own attribute names in order to avoid any conflicts between internal and custom attributes. Non-prefixed names are reserved for 4D Write Pro internal attributes. You can use any custom prefix (we used "myatt_" as a prefix in the above example).

📄 フォーム内の4D Write Proエリアを印刷する

4D v15 R3以降、4Dフォームに埋め込まれた4D Write Proエリアの印刷がサポートされるようになりました。

どのような4Dフォーム(プロジェクトフォーム、テーブルフォーム、入力/出力フォーム)でも、**PRINT SELECTION**や**PRINT RECORD**などの標準の4D印刷コマンドを使用して、そのフォームに含まれる4D Write Proエリアを印刷できるようになりました。

印刷時可変のオプションもサポートされます(*)。これにより印刷時にサイズの変化を管理できるようになりました。

(*)**Print object**と**Print form**コマンドはこのオプションをサポートしません。

固定長あるいは可変長で印刷

4D Write Proエリアの中身は、レコードごとにサイズが異なります。そのためエリアを固定長で印刷したり、それぞれのレコードに応じたサイズで印刷したりする必要があるでしょう。

4D Write Proフォームオブジェクトはプロパティリストの"印刷"テーマ内にある**印刷時可変**オプションをサポートしていません:



このプロパティは**OBJECT SET PRINT VARIABLE FRAME**と**OBJECT GET PRINT VARIABLE FRAME**コマンドを通して使用可能です。

- **固定長で印刷(印刷時可変オプションのチェックを外す):**

この設定では、フォームオブジェクトプロパティ(の"座標とサイズ"テーマ)で定義されている、固定の幅と高さで印刷されます。4D Write Proドキュメントの幅と高さはフォームオブジェクトのサイズに固定されます。ドキュメントの中身は必要に応じて折り返されます。ドキュメントの中身がエリアのサイズを超えた場合、超えた部分は印刷時に切り捨てられ印刷されません。

- **可変長で印刷(印刷時可変オプションをチェックする):**

この設定では、印刷プロセス時にエリアのサイズはドキュメントのサイズに応じて変化します。エリアは必要に応じて自動的に拡張され、一つまたは複数のページに中身が全て印刷されるようにします。4D Write Proドキュメントの幅はフォームオブジェクトプロパティで定義されたオブジェクトの幅に固定されますが、ドキュメントの高さはページ毎に調整され、中身のテキストが切り捨てられないようにします。ドキュメントの上マージン、パッド、境界線は最初のページにのみ適用され、ドキュメントの下マージン、パッド、境界線は最後のページにのみ適用されます。ドキュメントのページ区切りは無視されます。ウィドウ・オルファンコントロールは無効化され、"改ページ"プロパティは適用されません(これらのプロパティはスクリーン上のページレンダリング、あるいはスタンドアロンの印刷においてのみ適用されます)。印刷時可変オプションがチェックされている場合、エリアより上にあるオブジェクトのみが印刷されます。

例題

以下の例は、デフォルトの出力フォームに埋め込まれた4D Write Proエリアの**印刷時可変**オプションをチェックしたときとしないときの変化を示しています。以下のコードを実行します:

```
ALL RECORDS ([Movies])
ORDER BY ([Movies]Title)
PRINT SELECTION ([Movies])
```

- こちらが**印刷時可変**オプションの**チェックを外した(オフ)**時の結果です:

A Walk in the Woods

A Walk in the Woods



Avatar

Avatar



Black Mass



E.T. the Extra-Terrestrial

E.T. the Extra-Terrestrial



- こちらの印刷時可変オプションをチェックした(オン)時の結果です:

A Walk in the Woods



After spending two decades in England, Bill Bryson returns to the U.S., where he decides the best way to connect with his homeland is to hike the Appalachian Trail with one of his oldest friends.

Director: [Ken Kwapis](#)

Writers: [Rick Korb](#) (screenplay), [Bill Holderman](#) (screenplay), [2 more credits](#) »

Stars:

- [Robert Redford](#),
- [Nick Nolte](#),
- [Emma Thompson](#)
- [See full cast and crew](#)

Avatar



Avatar (marketed as James Cameron's *Avatar*) is a 2009 American^[1] epic science fiction film directed, written, produced, and co-edited by James Cameron, and starring Sam Worthington, Zoe Saldana, Stephen Lang, Michelle Rodriguez, and Sigourney Weaver. The film is set in the mid-22nd century, when humans are colonizing Pandora, a lush habitable moon of a gas giant in the Alpha Centauri star system, in order to mine the mineral unobtainium,^[2] a room-temperature superconductor.^[3] The expansion of the mining colony threatens the continued existence of a local tribe of Na'vi – a humanoid species indigenous to Pandora. The film's title refers to a genetically engineered Na'vi body with the mind of a remotely located human that is used to interact with the natives of Pandora.^[4]

Development of *Avatar* began in 1994, when Cameron wrote an 80-page treatment for the film.^[13]^[14] Filming was supposed to take place after the completion of Cameron's 1997 film *Titanic*, for a planned release in 1999,^[15] but according to Cameron, the necessary technology was not yet available to achieve his vision of the film.^[16] Work on the language of the film's extraterrestrial beings began in 2005, and Cameron began developing the screenplay and fictional universe in early 2006.^[17]^[18] *Avatar* was officially budgeted at \$237 million.^[3] Other estimates put the cost between \$280 million and \$310 million for production and at \$150 million for promotion.^[19]^[20]^[21] The film made extensive use of new motion capture filming techniques,^[22] and was released for traditional viewing, 3D viewing (using the RealD 3D, Dolby 3D, XpanD 3D, and IMAX 3D formats), and for "4D" experiences in select South Korean theaters.^[23] The stereoscopic filmmaking was touted as a breakthrough in cinematic technology.^[24]

(サンプルテキストソース: *Wikipedia*)

最適化

 Replace string

Replace string

新しい内部アルゴリズムのおかげで、4D v15 R3で**Replace string**コマンドを使用してある文字列を異なる長さの文字列で置き換える際の実行速度がとて速くなりました。これは以下のような置換を行う場合が当てはまります:

```
vResult:=Replace string(Source_Text;"a";"aa") //文字で置き換え  
vResult2:=Replace string(Source_Text2;"à";"aa";*) //文字コードで置き換え
```

新しいアルゴリズムはどちらのシンタックスにおいても最適化されています。ソースとなるテキストが長く、置換箇所が多いほど、この最適化の差は顕著に現れます。

私たちが行ったベンチマークでは、以前のアルゴリズムと比較して以下のような結果を得ました:

文字コードによる置き換え(* を渡した場合)	文字による置き換え(* 省略時)
------------------------	------------------

950 倍の速さ	4400 倍の速さ
----------	-----------




このテストはファイル内において32,000箇所の"a"を"aa"に置き換えるという内容で行われました。

注: 同じ長さの文字列で置き換える場合は、以前のアルゴリズムと同じ速さになります。

4D Developer Edition 64-bit版(プレビュー版)

4D v15 R2 より、4DはWindowsとOS X用の4D Developer Edition 64-bit版のオペレーショナルプレビュー版を提供し始めました。このバージョンについては、[4D v15 R2 - Upgrade \(PDF\)](#) のマニュアルに説明があります。

このバージョンの開発は、継続的なステップバイステップ方式で行われて来ました。4D Developer Edition 64-bit版のR3プレビュー版では、以前はご利用いただけなかった機能が有効化されています。再デザインされた機能に関する追加の情報も提供されています。

-  機能の状況
-  新設計されたプロパティリスト
-  新しいラベルウィザード(プレビュー版)

リマインダー: 4D Developer Edition 64-bit版の詳細な情報については、4D v15 R2アップグレードマニュアル(PDF)を参照して下さい。

更新された機能

多くの4D機能とダイアログが、64-bitアーキテクチャーをサポートするために更新、あるいは書き換えられました。ほとんどの変化は気づかれないように行われており、32-bit版のリリースと全く同じように動作します。しかしながら、一部のエディターにおいては変更が行われ、32-bit版とは違っています。これらは64-bit版4D Developer Editionにて提供されています。

機能	対象となる4Dのバージョン	補足
クイックレポートエディター	OS X & Win	完全に書き換えられています。カレントのプレビューでは、埋め込みエリアとクロステーブルレポートはご利用いただけません。プレビューについては 新しいクイックレポートエディター(プレビュー版) の章で説明があります。
ラベルエディター	OS X & Win	完全に書き換えられています。 新しいラベルウィザード(プレビュー版) の章を参照して下さい。
フォームエディタープロパティリスト	OS X & Win	新規デザイン

無効化された機能

最初の64-bit版4D Developer Editionでは、一部の機能は無効化されています:

機能/テクノロジー	対象となる4Dのバージョン	補足
シリアルポート通信	OS X	無効化されているコマンドは、 GET SERIAL PORT MAPPING 、[#cmd id="172"/]、 RECEIVE PACKET 、 RECEIVE RECORD 、 RECEIVE VARIABLE 、[#cmd id="103"/]、 SEND RECORD 、 SEND VARIABLE 、 SET CHANNEL 、[#cmd id="268"/]、それに加え、シリアルポートのコンテキストにおいてのみ USE CHARACTER SET も無効化されています。
読み込み/書き出しダイアログボックス	OS X	無効化されています。それに加え、 IMPORT DATA と EXPORT DATA も無効化されています。
クイックレポートの埋め込みエリア	OS X & Win	無効化
クイックレポートのクロステーブル	OS X & Win	無効化
Webエリア内でWeb Kitの統合	OS X & Win	無効化
チャート	OS X & Win	無効化(GRAPHコマンドはXSLTに依存するもので、これはサポートされていません)
印刷	Win	無効化

サポートされない機能

以下の機能またはテクノロジーは、64-bit版4D Developer Editionではサポートされません:

機能/テクノロジー	対象となる4Dのバージョン	コメント
XSLT と Xalan	OS X & Win	_o_XSLT APPLY TRANSFORMATION 、 _o_XSLT SET PARAMETER 、そして _o_XSLT GET ERROR は何もしません。代わりに PROCESS 4D TAGS コマンドを使用するか、PHP <i>libxslt</i> モジュールを使用して下さい。
PICT フォーマット	OS X & Win	'サポートされていない画像フォーマットです'ピクチャ+画像の拡張子が代わりに表示されます。PICTフォーマットは4D全体において廃止予定となっています。 AP Is Picture Deprecated も参照して下さい。
cicn アイコン	OS X & Win	GET ICON RESOURCE はサポートされていません。エラーを返します。
データベース .RSR ファイル	OS X & Win	データベース .RSR ファイルは自動的に開かれませんが、 Open resource file を使用する必要があります。
書き込み可能リソースファイル	OS X & Win	_o_Create resource file はサポートされていません。リソースファイルは読み込みのみモードでしか開く事ができません。
_o_Font number	OS X & Win	このコマンドはサポートされていません。これはエラーを返します。
旧式ネットワークレイヤー	OS X & Win	<i>ServerNet</i> のみがサポートされています。
ASCII 互換モード	OS X & Win	Unicode モードのみがサポートされています。
4D Write と 4D View プラグイン	Win	旧式プラグインはWindows用4Dの64-bit版では互換性がありません。代わりに4D Write Pro と4D View Pro(開発中)を使用して下さい。
OLE ツール	Win	サポートされていません

📄 新設計されたプロパティリスト

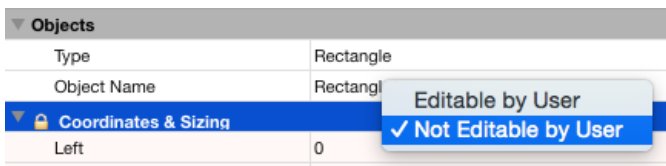
フォームエディターのプロパティリストは、4D Developer Editionの64-bit版において新設計されました。

これに付随して、ユーザーフォームエディターでロック/アンロックされたプロパティを設定・閲覧するために使用されていた機能が変更されました。

ユーザーフォームプロパティ

フォームが"ユーザー更新可"に設定されているとき、プロパティリストにおいて二つの新しい機能が使用できるようになりました:

- デベロッパは新しいコンテキストメニューを使用して編集可能なプロパティを選択できるようになりました:

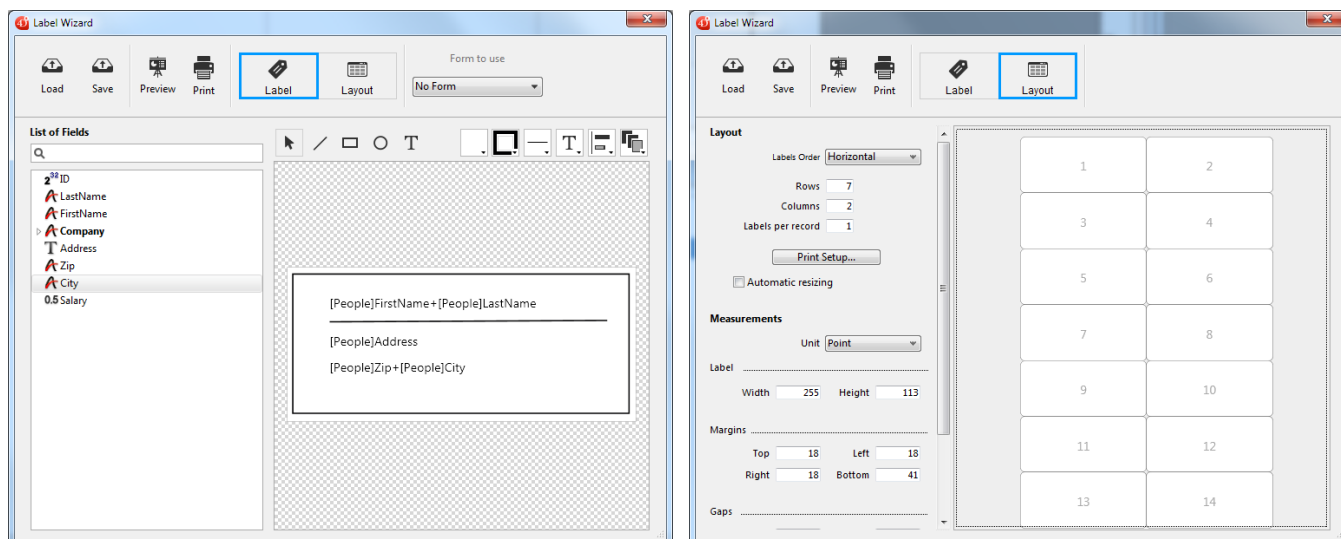


- 南京錠のアイコンに加えて、プロパティの背景色もロック・アンロックの状態を知らせるようになりました:
 - 薄い赤:ロック
 - 薄い青:アンロック



新しいラベルウィザード(プレビュー版)

64-bit版の4D v15 R2では新しく書き直された4D ラベルウィザードの製作中のプレビュー版が含まれています。是非時間を取ってこの新しいウィザードのインターフェースとカレントの機能をお確かめ下さい。以前のものに近いという事がお分かりいただけるかと思えます。これは将来の全てのバージョンの4Dにおいて現在の古いラベルエディターを置き換える予定です。



新しいラベルウィザードの主な特徴は以下の通りです:

- 互換性:
 - 以前のリリースで作成されたラベルファイルと互換性があります。フォーマットを変える事無く、以前のラベルを読み込み、編集、そして保存する事ができます。
 - 旧式のウィザードで利用可能だった全ての機能をサポートしています。
- 新しいコンテキストインターフェース:
 - ラベルページのグラフィック機能にアクセスするための、Ribbonベースのインターフェースとコンテキストメニュー。
 - "使用するフォーム"ポップアップメニューには使用可能なフォームのみが表示されます(使用可能とは、フォームがカレントテーブルに属していて、単一ページを含み、固定サイズウィンドウで表示されていると言う事が条件になります)。使用可能なフォームがデータベース内に無いときには、このメニューは表示されません。
 - プロジェクトにJSONファイルを追加する事によって使用可能なフォームとメソッドの一覧を定義する事ができます(以下を参照の事)。

注: の章に記載があるように、Windows用のこのプレビュー版では印刷はご利用いただけません。

互換性

ファイルフォーマット

新しいウィザードによって保存される4Dラベルのファイル拡張子は、".4lbp"となります。これは内部的にはXMLで書かれているので、このフォーマットはオープンであるということに注意して下さい。

以前のウィザードによって作成された古い4Dラベルファイル(".4lb"拡張子)はサポートされます。ウィザードはラベルのフォーマットを変える事無く、読み出し、変更、保存をすることができます。

注: 以前のラベルファイルでは、ピクチャーはPICTフォーマットで保存されていました。このフォーマットは4Dではサポートされなくなっているので(以下参照)、新しいエディターではこれらのピクチャーは赤い×印つきのエリアに置き換えられます。

サポートされない機能

以下の機能は、新しいラベルウィザードではサポートされません:

機能	コメント
フォーマットメニュー	String コマンドを使用して、例えば次のように書いて下さい: <pre>String([Emp]Birth;internal date long)</pre>
アウトラインスタイル	4Dではサポートされなくなりました
パターン	4Dではサポートされなくなりました
Quickdraw PICT	4Dでは廃止予定になりました /4D Developer Edition 64-bitではご利用いただけません
デフォルトリックボタン	有用ではなくなりました

無効化された機能

以下の機能は、新しいラベルウィザードでは現在無効化されています:

機能	コメント
標準コードポップアップメニュー	
Windowsでの印刷	4D Developer Edition 64-bit版(プレビュー)では無効化

新しい機能

使用可能なフォームとメソッドを定義する

新しいラベルウィザードでは、ダイアログで選択可能なフォームとメソッドをデータベースレベルで制限するという機能が新しく追加されました。以前のバージョンでは、デフォルトで、全ての利用可能なオブジェクトが表示されていました。

ラベルデザインにおいて選択可能なフォームまたはメソッドを定義するためには以下の手順に従って下さい:

1. **label.json** という名前のJSONファイルを作成し、データベースフォルダー内の**Resources** フォルダーにおきます。
2. このファイル内に、ラベルウィザードのメニュー内にて選択可能にしたいフォームまたはメソッド名を追加して行きます。

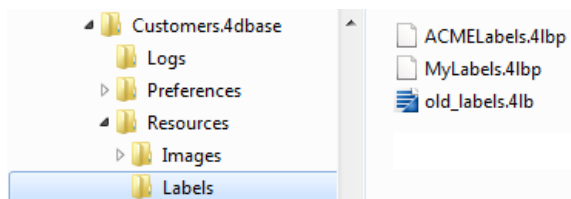
label.json ファイルの中身は、以下のようになっています:

```
[ { "tableId": 2, "forms": [], "methods": [ "myMethod1", "myMethod2" ] }, { "tableId": 1, "forms": [ "Sample Label 1", "Sample Label 2" ], "methods": [] } ]
```

ラベルファイルを事前に読み込む

新しいラベルウィザードでは、アプリケーション内にラベルファイルを保存することができます。これにより、**読み込み**ボタンを押す事によってユーザーが直接ラベルデザインを選択し開くことができるようになります。

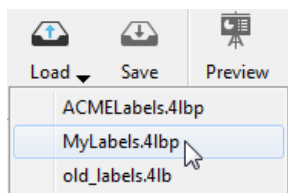
これをするためには、データベースの**Resource**フォルダ内に**Labels**という名前のフォルダを作成し、自分のラベルファイルをそこにコピーするだけです:



ラベルウィザードのスタート時、このフォルダが検出され、中に有効なラベルファイルがあることが確認されると、**読み込み**ボタンにポップアップアイコンが追加されます:

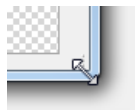


単一のメニューラインを使用してラベルデザインを選択する事ができます。新しい".4lbp"ファイルも、古い".4lb"もサポートされます:



ズーム

ウィンドウの右下のリサイズカーソルをドラッグすることによって、ラベルデザイン("Label"ページ)をズームする事ができます:



ドラッグ&ドロップ

"Labels"ページでは、ドラッグ&ドロップがサポートされています:

- デスクトップからラベルデザインエリアへとピクチャーファイルをドロップすることができます。
- デスクトップからラベルデザインエリアへとラベルファイル(".4ldp"ファイルのみ)をドロップすることができます。

向上した操作性

一般的な操作性が向上しました:

- スタティックテキストをラベルデザインエリアで直接入力できるようになりました。
- グラフィックオブジェクトはコンテキストメニューで管理できるようになりました
- 印刷プレビュー: 最初のページはラベルサイズを区切る透かしとともに印刷されます。