

4D v17への変換

"4D v17 への変換" マニュアルへようこそ。ここでは、4D v16 のデータベースを 4D v17用に
変換する前、変換中、そして変換後にチェックすべきさまざまな点についての説明をします。

4D v16 データベースの 4D v17 への変換はスムーズに行われるはずですが、すべてが上手く
行くためのいくつかの推奨点が "**変換の原則**" の章にまとめられています。しかしながら、変換
が終わった後にもいくつかチェックすべき事柄があります。それは "**新しい互換性オプション**"
と "**振る舞いの変更**" であり、どちらもアプリケーションレベルと 4D コマンドレベルで関わっ
てくるため、4D v17 の新機能を使いこなすにはどちらも理解しておく必要があります。

最後に、このマニュアルでは **4D v17 で廃止予定の機能**をまとめてあります。デベロッパーの
方はここを見ることで、機能の変更や代替の必要性を検討するための情報が手早く得られます。

注: このマニュアルに記載されている新機能・変更の一部は、4D v16の "R-リリース" プログラ
ムで既に紹介・導入されているものです。

以前の、あるいはさらにもっと古いデータベースを変換する際には、通常それらの間のバージョ
ンも介して変換することが必要になります。それぞれの変換の際にチェックすべきさまざまな項
目については、以前のバージョンでのドキュメントを参照してください:

- **4D v16:** "[4D v16への変換](#)" と "[廃止予定の機能と削除された機能 \(4D v16 以降\)](#)"
- **4D v15:** "[4D v15への変換 \(PDF\)](#)" と "[廃止予定の機能と削除された機能 \(4D v15 以
降・PDF\)](#)"
- **4D v14:** "[4D v14へのアップグレード \(PDF\)](#)" と "[廃止予定の機能と削除された機能 \(4D
v14 以降・PDF\)](#)"
- **4D v13:** "[4D v13へのアップグレード \(PDF\)](#)"
- **4D v12:** "[4D v12へのアップグレード \(PDF\)](#)"
- **4D v11:** "[4D v11 SQLへのアップグレード \(PDF\)](#)"

注: 各アップグレードマニュアルの最初の方で、前バージョンのデータベースの変換方法
と、廃止予定・削除済み機能について説明しています。

-  [変換の原則](#)
-  [互換性ダイアログ](#)
-  [振る舞いの変更](#)
-  [名前の変更、テーマの変更](#)

-  廃止予定の機能
-  無効化された機能
-  32-bitから64-bitへのアップグレード
-  4D Write から 4D Write Pro へドキュメントを変換
-  Convert 4D View documents to 4D View Pro
-  Appendix: 4D v16 Rx Release Notes
-  Appendix: Useful methods for conversion

変換の原則

変換の前にはしておくべきこと

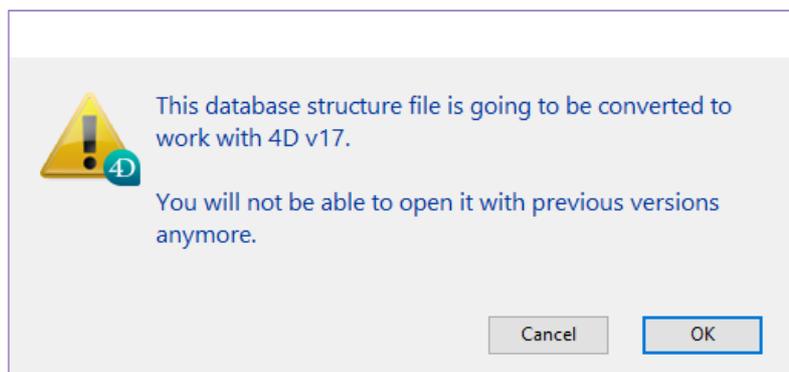
- 変換を行うためには、データベースの "**インタープリター**" 版 (ストラクチャーの xxxx.4DB ファイル、データの xxxx.4DD ファイル、あるいは 4D v11 より前のバージョンからの変換には .RSR および .4DR ファイル - 詳細は古いバージョンの "変換" ドキュメントを参照してください) と、**スーパーユーザー** パスワードが必要になります。
- 変換の前に、**必ずデータベースのコピー (バックアップ) を作成してください。**
- シNTAX チェックを実行します。データベースをコンパイルしない場合も、このチェックによって起こり得るエラーを検知することができます。
- **Maintenance and Security Center (MSC)** を使用してストラクチャーとデータの検査と修復を行ってください。
- **GET PICTURE FORMATS** コマンド を使用して、データベース内に PICT ファイルがあるかどうかをチェックし、あった場合には **CONVERT PICTURE** コマンドを使用して変換します
4D v14 では、デフォルトでは QuickTime コーデックはサポートされていません。**SET DATABASE PARAMETER** コマンドの **QuickTime Support** セレクターを使用することはできますが、このオプションの変更にはデータベースの再起動が必要です。ただし、QuickTime のサポートは、将来のバージョンの 4D においては恒久的に削除されるという点に注意してください。
廃止予定の画像フォーマットを変換するためには、**Conversion of picture type** を参照して下さい。
- (任意) データのジャーナリングが必要な場合、プライマリーキー (v14 以降実装) を実装することができます (デザインリファレンス マニュアルの **主キーを設定、削除する** を参照ください)。プライマリーキーを設定することは強く推奨されていますが、これは変換の後でも設定することが可能です。
- v13.5 以降のデータベースにおいて、**重複不可** 属性を持つフィールドは **インデックスが必須** となりました。今後はインデックスがついていない重複不可属性のフィールド内では一切レコードを作成/編集することはできなくなります。レコードを保存しようとする、エラーが生成されます (-9998 重複不可のレコードが存在します・1088 インデックスが無効または未設定です)。存在しないインデックスの作成方法、またはインデックスがなされていないフィールドをすべてまとめたディスクファイルの生成方法については、**To create missing indexes** を参照してください。

変換の方法

v16 の 4D または 4D Server (v11、v12、v13、v14、v15 も同様) を使用して作成されたデータベースは 4D v17 において互換性があります (ストラクチャーとデータファイル)。どんなインタープリター版のストラクチャーファイルも変換することができます。変換するためには、4D v17 を起動し、そのストラクチャーファイル (xxx.4DB ファイル) をインタープリターモードで開くだけです。

• ストラクチャーファイルの変換(.4DB)

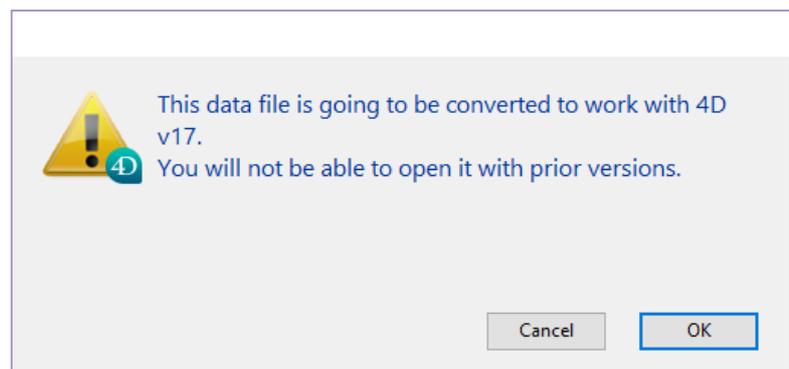
ストラクチャーファイルが変換されることを警告するダイアログが表示され、バージョンによってはデータファイルの変換のダイアログも表示されます:



ストラクチャーファイルは4D v17 へと変換され、それより前のバージョンでは開くことはできなくなります。

• データファイルの変換(.4DD)

4D v14 およびそれ以前のデータファイルは変換される必要があります。この場合には、二つ目のダイアログが表示されます:

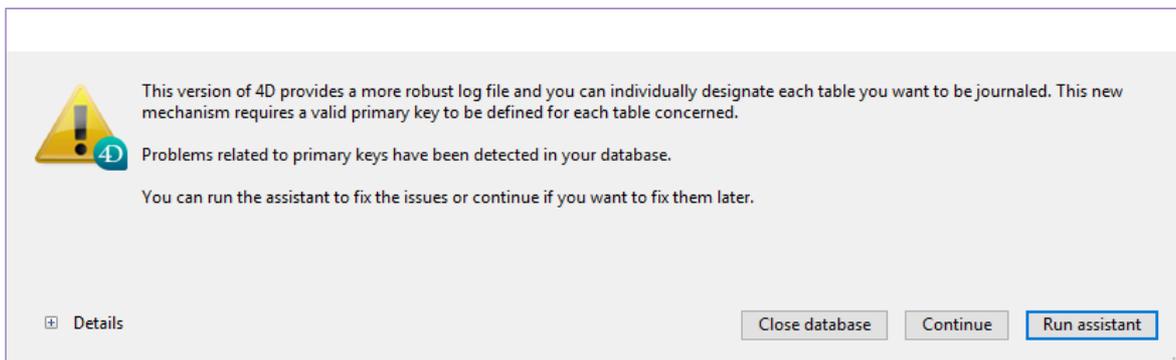


データファイルはv17 へと変換されますが、4D v15 と4D v16 では引き続き開くことができます。

4D v15 および4D v16 のデータベースではデータを変換する必要はありません。

• ログファイル設定ダイアログ

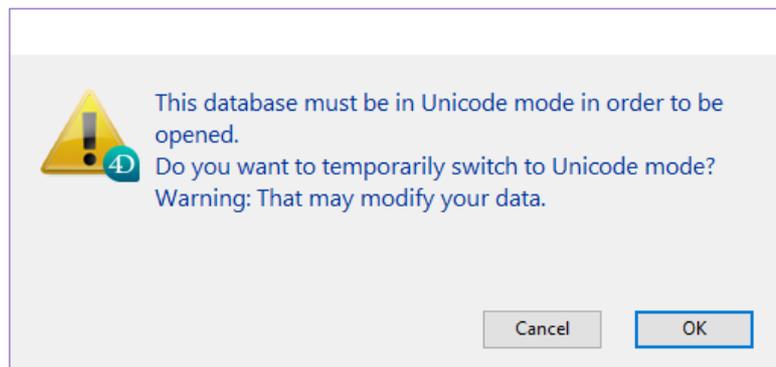
プライマリーキーが設定されていない場合、4D は以下のダイアログを表示して設定を促します:



プライマリーキーは、(バックアップのために)ログを必要とするテーブルに対して設定する("ウィザードを使用"ボタンを押す)ことが望ましいですが、これは後ほど設定する("続ける"ボタンを押す)こともできます。詳細な情報については、[プライマリーキー管理](#) の章を参照して下さい。

• 一時的Unicode ダイアログ

アプリケーションを64-bit 版の4D v17 で開き、そのアプリケーションがUnicodeモードでない場合、4D はデータベースを一時的にUnicode に切り替えるかどうかを聞いてきます。



Unicode ではスピードが速くなるため、ここ最近のバージョンにおいては切り替えることが推奨されてきました。まだ切り替えていない場合には、ただちに切り替えましょう。詳細な情報については、[互換性ページ](#) の章を参照して下さい。

変換の後で

Maintenance and Security Center(MSC) を再度使用してストラクチャーとデータの検査と修復を行ってください。

ストラクチャーにおける留意点は以下の通りです：

- オーフアンメソッド (__Orphan__xxxx) は MSC のログファイルに**警告**として記録されます。オーファンメソッドのコードが不要であることを確認したあと、これらはエクスプローラーを使って削除することができます。
- フォーム内でのオブジェクト名の重複は許可されていません。これらはMSC のログファイルに**警告**として報告されます。データベースの修復を実行し、これらの名前を変更して下さい(このとき、オブジェクト名のプログラミングに注意するようにして下さい)。

- 廃止予定の画像(PICT フォーマット)の検知。MSC の章の**アプリケーションの検証** を使用して下さい。

廃止予定または削除済み機能の一覧マニュアル内の**PICT フォーマットのピクチャー** の段落を参照して下さい。これらの警告はスタティックな画像と、画像ライブラリあるいはフォームオブジェクトに保存された画像に影響する可能性があります。

注: これらの画像の変換には、**Conversion of picture type** を参照して下さい。

- **名前の中の不適切な文字 (".", "[", および "]")**

4D v16 R4 以降、以下の要素においてピリオド(.) および大カッコ([]) を使用することは推奨されていません:

- 変数名
- テーブル名
- フィールド名
- プロジェクトメソッド名

デベロッパーが、アプリケーションをこれらの規則を遵守させるのを容易にするために、**アプリケーションの検証** アクションを使用すると、変数名、テーブル名、フィールド名、メソッド名の中にあるこれらの不適切な文字の存在を自動的に検知します。これらの文字が検知された場合、MSC によって"禁則文"が報告され、ログファイルには適切な警告が含まれます:

Check Tables And Fields Names

Warning: The Table "Table.1" contains dots or square brackets(0;0)

こういった場合、アプリケーション内のこれらの要素を改名することが強く推奨されます。

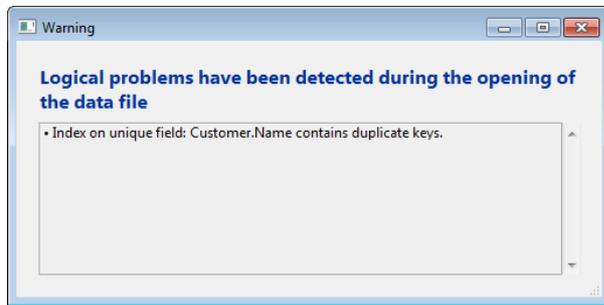
データ情報について:

- 重複不可フィールドでの重複の検知:

追加の情報が提供されます: MSC、あるいは**VERIFY DATA FILE** のようなコマンドを使用した場合、ログファイルには関係するテーブル名とフィールド名にくわえて、重複している値も含まれます。

注: データ入力時、"キー重複"エラーダイアログには関係するテーブル名とフィールド名、それから重複した値が含まれます。また**GET LAST ERROR STACK** コマンドも可能性のある重複について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、インデックスを構築(あるいは再構築)する必要がある場合、割り当てられた宣言されたフィールド内での値の重複は自動的に検知されるようになりました。この場合、データベースを開く前に特定の警告ダイアログボックスが開かれ、重複を見つけて削除するための情報をユーザーに提供します:



インデックスの再構築

v16 から v17 への変換では、日本語版の4D のみデータベースインデックスの再構築が必要です。

また、古いバージョンからのアップデートの場合(特に、Unicode ライブラリ - ICU - International Components for Unicode - のアップデートが必要な場合)、4D 内の全てのテキストとキーワードインデックスは再構築することができます。この処理はデータベースの初めて開いたときに自動的に行われます(注意: この処理にはかなりの時間を要する場合があります)。

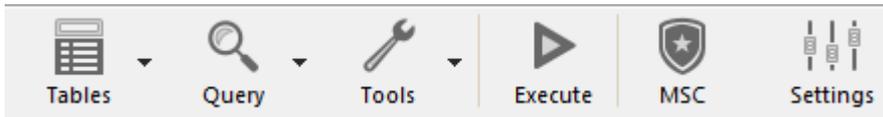
注: 4D v16 以降データベースデータ全体の再インデックスのアルゴリズムが大幅に最適化されました。プロセス全体を見直すことで、最大で2倍の速さを達成しました。全体再インデックスは、例えばデータベース修復後や.4dindxファイルが削除された後などでは必須です。

互換性に関する注意

4D v16.x のデータベースを、v16.Rx で開くことやその逆も可能です。しかしながら、新機能を使用したコードは動作しませんし、無効化されるべきです。v17 に変換してしまったデータベースはv16 で開くことはできなくなります。

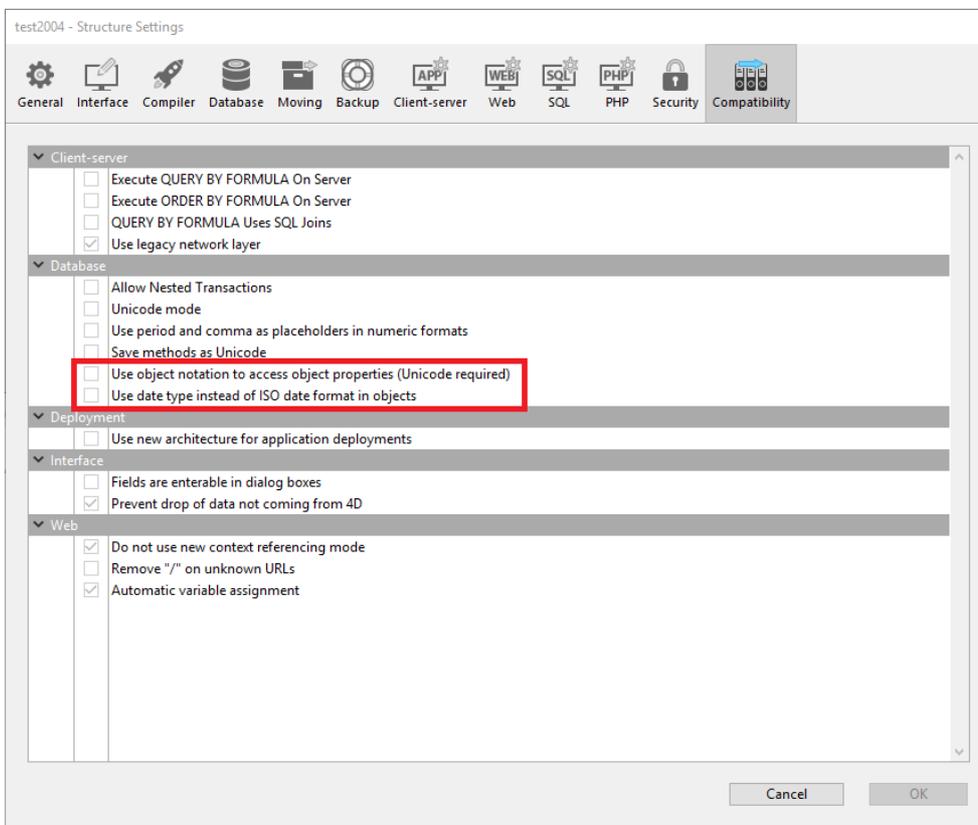
互換性ダイアログ

このダイアログに行くためには、まずメインのツールバーの "設定" をクリックします:



その後 "互換性" のタブをクリックします。

v17 では二つの新しいオプションが追加されました。



オブジェクト記法を使用してオブジェクトプロパティにアクセス

オブジェクト記法([オブジェクト記法の使用参照](#))を利用したい場合、このオプションは ":", "[" および "]" といった文字をオブジェクト記法シンボルとして使用する事で、トークナイズされたプロパティにアクセスすることができるようになります。使える場所は**変数名**、**テーブル名**、**フィールド名**、および**プロジェクトメソッド名**です。

4D v17 より前のバージョンで作成されたデータベースにおいては、このオプションの有効化は必須です。以前のバージョンでは ":", "[" および "]" といった文字を要素の名前として使用する事が許可されていましたが、今後は使用できなくなるからです。

警告 :

- このオプションを使用するためには、データベースがUnicode モードになっている必要があります。
- このオプションを有効化する際、MSC を使用して既存のコードが命名規則を遵守しているかどうかをチェックする事が推奨されます([アプリケーションの検証](#) 参照)。デベロッパーがアプリケーションをこのルールに合致させるのを容易にするために、**アプリケーションを検証**アクションは変数名、テーブル名、フィールド名、メソッド名の中にあるこれらの不要な文字の存在を自動的に検知します。これらの文字が検知された場合、MSC によってエラーが報告されます:

Check Tables And Fields Names

Warning: The Table "Table.1" contains dots or square brackets(0;0)

この場合、アプリケーション内にてこれらの要素の名前を変更しなければなりません。

オブジェクト内でISO日付フォーマットの代わりに日付型を使用

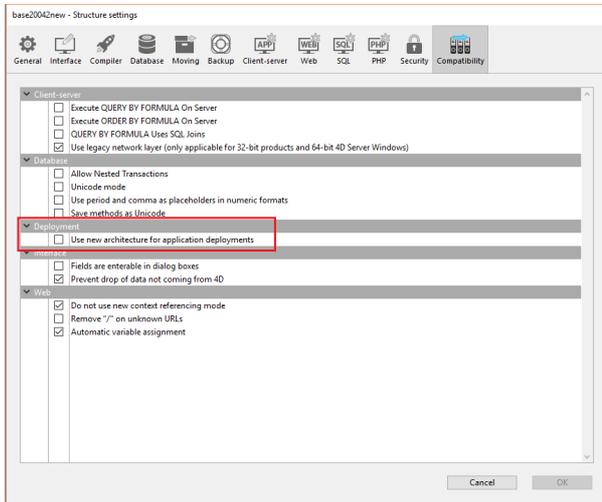
オブジェクト内にて、日付を文字列ではなく日付として定義できるようになりました。これにより、オブジェクト内での日付型の使用がより簡単で直感的になりました。これにより、**OB SET** および **OB Get** コマンドは `Is date` 定数を使用する事なく使えるようになり、他のオブジェクト同様にオブジェクト記法を適用する事もできます。変換されたデータベースにおいてオブジェクト内で日付型を使用するには、"オブジェクト内でISO日付フォーマットの代わりに日付型を使用"オプションをチェックするだけです。[互換性ページ](#)を参照してください。

削除されたオプション

"ブラケットの代わりに4DVARコメントを使用する" オプションは 4D v16 の[互換性](#)ページより削除されました。

古い互換性オプション

変換元の4D のバージョンが古ければ古いほど、より多くのオプションが表示されます:



詳細な情報については、[互換性ページ](#)を参照してください。

振る舞いの変更

Objet Get action

OBJECT Get action および **OBJECT SET ACTION** コマンドは、フォームオブジェクトに対して使用する際、倍長整数の定数ではなく文字列の定数を使用するようになりました。

以前のコード:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(OBJECT Get action(*;$arrObjects{$i})=Object No standard action)
        OBJECT SET ACTION(*;$arrObjects{$i};Object Cancel action)
    End if
End for
```

v17 に変換後のコード:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(_o_OBJECT Get action(*;$arrObjects{$i})=_o_Object No standard action)
        OBJECT SET ACTION(*;$arrObjects{$i};_o_Object Cancel action)
    End if
End for
```

注: _o_Object Get action ファンクションのみが廃止予定と宣言されています。なぜならコンパイル時、結果の型をあらかじめ指定する必要があるからです。これはユーザーにコードの書き換えを促すために実装されました(以下参照)。そうでなければ、コンパイルは不可能であったからです。

新コード: コマンドや廃止予定の定数は以下のように置き換えることができます:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(OBJECT Get action(*;$arrObjects{$i})=ak none)
        OBJECT SET ACTION(*;$arrObjects{$i};ak cancel)
    End if
End for
```

時間は秒数での表現に

v17 以降、オブジェクトプロパティによって操作される4Dの時間(**C_TIME**型)は**秒数**へと変換されます。以前のバージョンでは、これはミリ秒へと変換されていました。

この変化は、オブジェクト記法を用いてオブジェクトあるいはコレクションに(あるいはそれから)変換されるすべての時間に適応されます。**OB SET** および **OB Get, QUERY BY ATTRIBUTE** といったコマンド、あるいは**JSON Stringify** および **JSON Parse** といった JSON コマンドなどです。また、以下の4D 機能における時間/数値の変換にも影響します:

- Web エリア(JavaScript 経由)
- 4D Mobile
- SQL (**CAST** ファンクション)

4D データベース側の適応を容易にするため(特に時間がフィールドオブジェクト属性内に保存されていたような場合)、新しい互換性オプションを使用して、セッション中に以前の振る舞いに戻すことができます:

```
//以前の振る舞いに戻す
// On opening / On server startup データベースメソッドで実行されることを想定
SET DATABASE PARAMETER(Times inside objects;Times in milliseconds)
```

XMLの編集

• 外部エンティティ解決

4D v16R3 以降、外部エンティティの解決はXML ドキュメントではデフォルトでは有効化されていません(外部エンティティの宣言は解析エラーを生成します)。以前の振る舞いに戻すためには、**XML SET OPTIONS** コマンドにおいて、新しいXML external entity resolution セレクターとXML enabled 値を使用します。

• 大文字と小文字の区別

4D v16 R4 以降、**DOM Get XML element** および **DOM Count XML elements** コマンドはデフォルトでは大文字と小文字を区別するようになりました。新しいセレクターを使用すると、以前の振る舞いに戻すことができます: XML DOM case sensitivity では以下の値を使用できます:

- XML case sensitive (デフォルトの値): コマンドは大文字と小文字を区別します。
- XML case insensitive: コマンドは大文字と小文字を区別しません。

4D Write Pro および画像属性

WP GET ATTRIBUTES あるいは**OB Get** コマンドでwk image, wk list style image, および wk background image 属性を使用すると、画像が宣言されていない場合、画像のURL ではなく、画像そのものを返すようになりました。

画像URL を探すためには、新しい wk image url, wk list style image url, および wk background image url 属性を使用してください。

名前の変更、テーマの変更

4D Write Pro

- WP Get は**WP Create paragraph range** へと改名されました。
- WP Get pictures は**WP Create picture range** へと改名されました。

標準アクション定数

"標準アクション" テーマの数値定数は廃止予定("_o_" の接頭辞付き)となっています。4D v16 R3 以降では、"ak ..." の接頭辞がついたテキスト定数を使用することが推奨されます。

廃止予定の機能

廃止予定のコマンド

データベース内で使用されている廃止予定のコマンドを探すには、編集メニューから**デザインモードを検索**を選択するか、ツールバーの**デザインモードを検索**エリアを使って、廃止予定コマンドの接頭辞である "_o_" を検索します。廃止予定のコマンドは新しいコマンドや機能に代替することが強く推奨されますが、"削除" されていないかぎり、これらの廃止予定コマンドは機能することにも留意してください。

廃止予定コマンドの完全な一覧については、[ランゲージ: 廃止予定または削除されたコマンド](#)の章を参照してください。

4D v16で廃止予定となったコマンド

- 4D コマンド

以前の名前	新しい名前	コメント	廃止予定になった時期	現在の状況
Object Get action	OBJECT Get action (または変換の場合は _o_OBJECT Get action)、ただし引数に変更あり (倍調整数値の引数から文字列値の引数へと変更)	変換すると、 _o_Object read action コマンドは振る舞いに変更がありません ("振る舞いの変更"の章を参照)。	v16R3	廃止予定
PICTURE TO GIF	_o_PICTURE TO GIF	PICTURE TO BLOB	v16R5	廃止予定
Type document	_o_Document type	Path to object コマンドを使用のこと	v16R6	廃止予定
Document creator	_o_Document creator	Path to object コマンドを使用のこと	v16R6	廃止予定
SET DOCUMENT TYPE	_o_Document type	Object to path コマンドを使用のこと	v16R6	廃止予定
SET DOCUMENT CREATOR	_o_SET DOCUMENT CREATOR	Object to path コマンドを使用のこと	v16R6	廃止予定
MAP FILE TYPES	_o_MAP FILE TYPES	UTI や info.plist を使用のこと	v16R6	廃止予定
Gestalt	_o_Gestalt	Get system info / Is macOS / Is Windows によって置き換え	v17	廃止予定
PLATFORM PROPERTIES	_o_PLATFORM PROPERTIES	Get system info / Is macOS / Is Windows によって置き換え	v17	廃止予定

- 4D 定数

以前の名前	新しい名前	コメント	廃止予定になった時期	現在の状況
<u>lk</u> <u>affichage</u> <u>barre déf</u> <u>hor</u>	<u>_o lk</u> <u>display</u> <u>hor</u> <u>scrollbar</u>	OBJECT GET SCROLLBAR	v16R3	廃止予定
<u>lk</u> <u>affichage</u> <u>barre déf</u> <u>ver</u>	<u>_o lk</u> <u>display</u> <u>ver</u> <u>scrollbar</u>	OBJECT GET SCROLLBAR	v16R3	廃止予定
<u>lk</u> <u>position</u> <u>barre déf</u> <u>hor</u>	<u>_o lk hor</u> <u>scrollbar</u> <u>position</u>	OBJECT GET SCROLL POSITION	v16R3	廃止予定
<u>lk</u> <u>position</u> <u>barre déf</u> <u>ver</u>	<u>_o lk ver</u> <u>scrollbar</u> <u>position</u>	OBJECT GET SCROLL POSITION	v16R3	廃止予定
<u>lk</u> <u>hauteur</u> <u>ped</u>	<u>_o lk</u> <u>footer</u> <u>height</u>	LISTBOX Get footers height	v16R3	廃止予定
<u>lk</u> <u>hauteur</u> <u>entête</u>	<u>_o lk</u> <u>header</u> <u>height</u>	LISTBOX Get headers height	v16R3	廃止予定
<u>_o Objet</u> <u>xxx</u>	<u>ak xxx</u>	標準アクションをサポートする定数。特に OBJECT Get action および OBJECT SET ACTION cコマンドを参照してください。	v16R3	廃止予定

- 4D Internet Commands

以前の名前	新しい名前	コメント	廃止予定になった時期	現在の状況
<i>FTP_Progress</i>	呼び出された場合にはエラーを返します	進捗パラメーターはFTP_Append, FTP_Receive, FTP_Send コマンドではサポートされなくなりました。	v16R3	無効化

無効化された機能

- **4D Pack** プラグインは、4D v16R2 以降削除されました。
注 : v16.x 用の4D Pack は4D v16 Rx と互換性があります。
- **FTP_Progress**: 4D Internet Commands の中で、このコマンドは4D v16 R2 以降無効化されました。今後は呼び出された場合にはエラー -2201 を返します。
注: 以下のコマンドの *progress* 引数はサポートされなくなりました:
FTP_Append
FTP_Receive
FTP_Send

📄 32-bitから64-bitへのアップグレード

4D v17 は32-bit 版でも64-bit 版でも、macOS でもWindows でもお使いいただけます。

32-bit版の4D アプリケーションを64-bit版へとアップグレードするには、多少の準備が必要となります。

64-bit 版4D プロダクトラインは全て、Altura のMac2Win ライブラリに依存しなくなりました。**Altura のMac2Win が64-bit 版の4D Developer Edition および4D Volume Desktop から完全に除去された事で、これらの製品は現代的なWindows APIを自由に使用する事ができるようになりました:**

- 64-bit 版 4D Developer Edition および4D Volume Desktop Windows のv17 (厳密にはv16 R2 以降)
- 64-bit 版 4D Server Windows v17 (厳密にはv16 R4 以降)

注: 32-bit 版のWindows 用製品では、4D は引き続きMac2Win を使用しています。

64-bit 版4D Developer Edition では、新しいラベルエディター、クイックレポート、読み込み/書き出しエディターなど、現代的で直感的で、使いやすい機能が統合されています。アプリケーションを複数回インストールする事なく、64-bit 版4D Developer Edition のインスタンスを複数起動する事ができます。

お使いのアプリケーションが64-bit 版のWindows 用あるいはmacOS 用の4D Server を使っているなら、移行のための作業はすでにほとんど完了しています。64-bit 版のシングルユーザー向けアプリケーションの場合は、いくつかの追加ステップが必要になるかもしれません。この章では、アップグレードの前と後に必要なステップをチェックするためのポイントをまとめたチェックリストを用意しています。

いくつかの機能はプロダクトの64-bit 版への移行に伴い、更新されたり、無効化されたり、あるいは廃止予定であると宣言されています。全ての詳細は、**64-bit版の 4D Developer Edition** の章の の段落にまとめてあります。

注: 通常のアップグレードのプロセス同様、重要なステップの前にはMSC を使用してチェックを実行し、データとストラクチャーが有効であることを確認していく事が推奨されます。

プラグインをチェックする

最初にすることは、プラグインがあれば、それらを64-bit版へとアップグレードすることです:

- **4D プラグイン:**
4D Write と4D Viewを除き、すべてのプラグインは既に64-bit版で動作しています。

- お使いのアプリケーションが4D Writeを使用している場合、そのコードを4D Write Proへと移行することを検討する必要があります。このとき、既存の32-bit版のコードはそのまま保持しておき、並行して新しい64-bitベースのモジュールを4D Write Proで作りはじめるのが良いでしょう。移行期間の間に、4D Write および4D Write Pro ドキュメントの両方を使用したい場合には、32-bit 版の4D v17 を使用するのが良いでしょう。
 - お使いのアプリケーションが4D Viewを使用する場合、4D View Proあるいは別の選択肢を使用する必要があります。
- **サードパーティーのプラグイン:**
プラグインのプロバイダーに連絡をして、64-bit版を取得してください。

32-bit版においてアップグレードに必要な準備

1. お使いのアプリケーションを最新の32-bitリリース (例えば4D v17 32-bitなど) にアップグレードしてください。
2. Unicodeモードが有効化されていることを確認してください。
3. あらゆるPICT/cicn/QuickTimeピクチャーを変換してください。
データ内にある廃止予定のピクチャー形式を検出するには、**GET PICTURE FORMATS** コマンドが使用できます。
また、データベースのストラクチャー内にあるサポートされていないピクチャーもすべて置き換える必要があります。MSCで検証をすることで、resources 内にあるピクチャーや3Dボタン用のピクチャーや静的なピクチャーの中で古いピクチャーを検知することができます。
4. 全てのXSLTベースの機能 (**_o_XSLT APPLY TRANSFORMATION**、**_o_XSLT SET PARAMETER** あるいは **_o_XSLT GET ERROR** コマンド)を、例えば **PROCESS 4D TAGS** コマンドで置き換えてください。
5. **_o_Font number** の呼び出しをフォント名での呼び出しに置き換えてください。
6. リソースファイルを作成あるいは変更するコードをすべて削除してください。

この時点で、お使いのアプリケーションを 64-bit版の4Dで開く準備ができました。

データベースを64-bit版で開いてチェックする

1. お使いのアプリケーションを 4D Developer Edition 64-bit版で開いてください。
2. Webエリアにおいて統合された WebKit を使用している場合、これらは自動的にシステムエンジンに切り替わっているため、チェックしてみてください (4Dメソッド(\$4d) へのアクセスは引き続き有効です)。
3. コード内で **SET PRINT OPTION** コマンドの Mac spool file format option を使用している場合、それらを **SET CURRENT PRINTER** と Generic PDF driver 定数の組み合わせの呼び出しで置き換える必要があります。
4. ラベルエディターの呼び出しと使用をチェックしてください (**ラベルエディター (新デザイン)** 参照)。

5. クイックレポートの呼び出しと使用をチェックしてください ([クイックレポートエディター \(新デザイン\)](#) 参照)。

これでお使いのアプリケーションは完全に64-bit版で使用できるようになり、4D 64-bit版の新機能も使用できるようになりました。

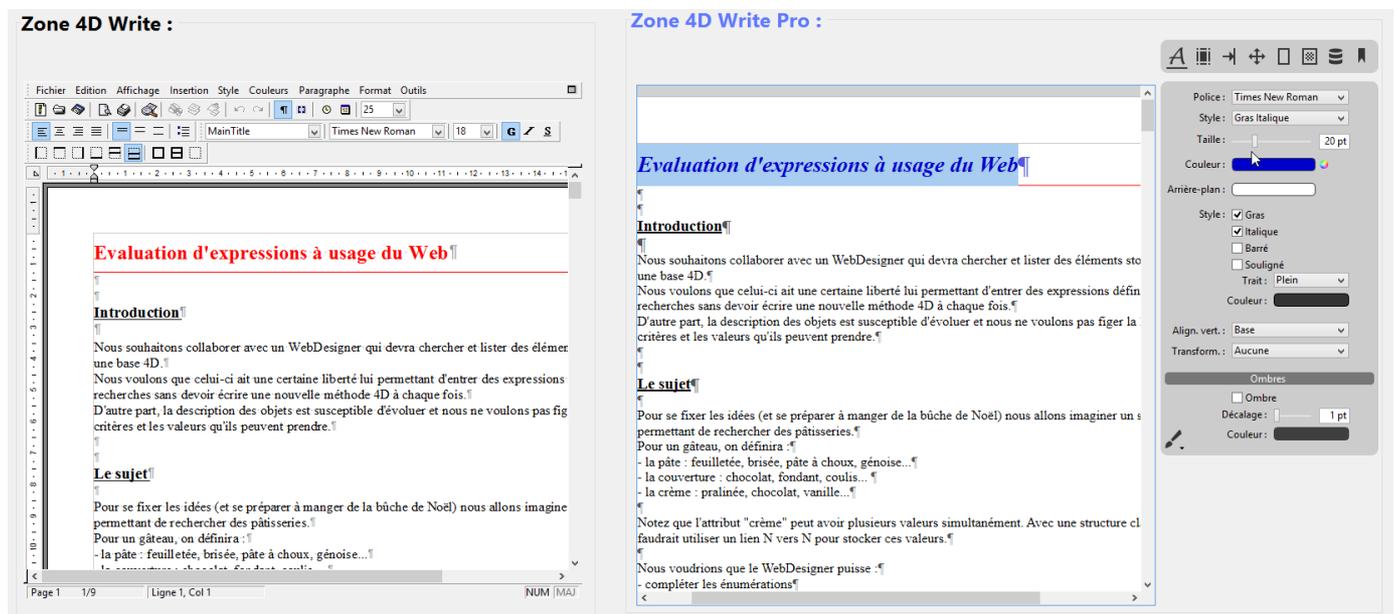
64-bit版の機能の利点

- **64-bitアーキテクチャー**はデータベースキャッシュの限度を引き上げます。より大きいキャッシュを使用するだけでデータベースのパフォーマンスを向上させることができます。プリエンティブプロセス、アニメーション付きフォームオブジェクト、新しい印刷機能など、**強力な64-bit機能**を使用してみましょう。
4D Runtime Volumeライセンス64-bit版でアプリケーションをビルドしましょう。
Win および macOS で 64-bit版 4D Server のファイナルバージョンを使用しましょう ([OS X用64ビット版4D Serverの使用について](#) および [Windows用4D Serverの64ビット版](#) 参照)。
- **新しいクイックレポートエディター** は旧版で作成したレポートにも対応しています ([ラベルエディター \(64-bit\)](#) 参照)。
- **新しいラベルエディター** は旧版で作成したラベルファイルにも対応しています ([クイックレポート\(64-bit版\)](#) 参照)。
- **GRAPH** コマンドにオブジェクト型パラメーターを指定して**グラフ**を作成しましょう。

4D Write から 4D Write Pro へドキュメントを変換

4D Write ドキュメントの変換

4D Write Pro は、ほとんどのプロパティ情報を残したまま、4D Writeドキュメントを開いて変換することができます。



上図では、左に 4D Write エリア、右に 4D Write Pro エリアを表示しています (新しいライブラリオブジェクトを使って作成 - 後述参照)。4D Write エリアの中身は **WP New** コマンドを使って簡単に変換することができます:

```
// 4D Write エリアのコンテンツを 4D Write Pro に復元します。  
[WRITEAREAS]AreaNTWP:=WP New([WRITEAREAS]AreaNT_)
```

ただし、4D Write は 32-bit版の 4D でのみ使用できることに注意が必要です。まず 4D Write ドキュメントを変換してから、64-bit に変換する必要があります。

4D Write と違い、4D Write Pro はプラグインではなく、4D に完全に統合されています。4D Write Pro には 4D Write と同じライセンスが使用されるので、4D Write Pro 機能を有効化するには、アプリケーションにこのライセンスをインストールする必要があります。

4D Write Proオブジェクトに 4D Writeドキュメントを取得する方法は二つあります:

- ディスク上に保存されている 4D Write ファイルに関しては、**WP Import document** コマンドを使用します。4D Write ファイル(.4w6、.4w7、および.4wt - テンプレート) は4D Write Proファイル(.4wp)へと変換される必要があります。

```
// まず、.4w6 ファイルを.4w7 ファイルへと、4D Write コマンドを使用して変換します。  
$offscreen:=WR New offscreen area  
WR OPEN DOCUMENT($offscreen;"myFile.4w6";"4WR6")  
WR SAVE DOCUMENT($offscreen;"myFile.4w7";"4WR7")  
WR DELETE OFFSCREEN AREA($offscreen)
```

```
// 次に.4w7 を .4wp へと、4D Write Proコマンドを使用して変換します。  
C_OBJECT($docWritePro)  
$docWritePro:=WP Import document("myFile.4w7")  
WP EXPORT DOCUMENT($docWritePro;"myFile.4wp")
```

- データファイルに格納されている 4D Write エリアに関しては、**WP New** コマンドを使用します。BLOBあるいはピクチャーフィールドに保存されている4D Write ファイルはオブジェクトフィールドへと移動させる必要があります。

```
// ピクチャーフィールドからBLOBを経由して、オブジェクトフィールドへと移動  
// [DocWRITE]WritePictArea_ はピクチャーフィールド  
// $Blob は BLOB  
// [DocWRITE]WriteProArea はオブジェクトフィールド  
$offscreen:=WR New offscreen area  
WR PICTURE TO AREA($offscreen;[DocWRITE]WritePictArea_)  
$Blob:=WR Area to blob($offscreen;1)  
[DocWRITE]WriteProArea :=WP New($Blob)  
WR DELETE OFFSCREEN AREA($offscreen)
```

```
// BLOB フィールドからオブジェクトフィールドへと移動  
// [DocWRITE]WriteBLOBArea_ は BLOB フィールド  
// [DocWRITE]WriteProArea はオブジェクトフィールド  
[DocWRITE]WriteProArea :=WP New([DocWRITE]WriteBLOBArea_)  
WR DELETE OFFSCREEN AREA($offscreen)
```

互換性に関する注意:

- サポートされるのは、4D Write ドキュメントのうち最後の世代 ("4D Write v7") のドキュメントに限られます。
- 4D Write Proエリアにインポートできるオブジェクトや機能についての詳細は **4D Write のプロパティで復元されるもの** を参照ください。

- 4D Write ドキュメントから 4D Write Pro エリアへのコピー・ペーストは現時点ではサポートされていません。4D Writeドキュメントの読み込みは 4D Write Pro ランゲージコマンドの使用によってのみ可能です。
- 4D Write Pro の機能は Windows において Direct2D に依存しています。必要となる Direct2D のバージョンを利用するため、Windows 7 または Windows Server 2008 のマシンでは "Windowsプラットフォーム更新プログラム" のコンポーネントがインストールされていることを確認してください。

4D式のフィルタリング

以前のバージョンにおける 4D Write Pro ドキュメント内では、フィルタリングが有効化されていませんでした。4Dメソッドを使用している 4D Write Pro ドキュメントを 4D v16以降へと変換した場合、これらは正しく評価されなくなり、代わりに "## Error # 48" メッセージが表示されます。

この場合、**SET ALLOWED METHODS** コマンドを使って、許可されたメソッドのリストに使用したいメソッドを追加する必要があります。

変更されたコマンド

4D Write Pro で使用できるように、新しいコマンドが追加され、古いコマンドは進化しました:

- 水平ルーラー: マージン、インデント、タブの調整に使用
- カスタムツールバー: 標準アクション機構を拡張
- **Dynamic pop up menu** コマンドの更新: 標準アクションに基づいた独自の4D Write Pro コンテキストメニューのデザインが可能に
- 表管理: **WP Insert table**, **WP Table append row**, **WP Table get rows**, **WP Table get columns**, **WP Table get cells**
- ハイパーリンク: **WP SET ATTRIBUTES** および **WP GET ATTRIBUTES** コマンドに対して、新しい `wk link url` 属性が追加されました。
- 画像挿入: **WP Add picture** コマンドと、**WP SET ATTRIBUTES** コマンドのフルページ背景画像が可能になりました(属性: `wk paper box`)。
- ヘッダーおよびフッター管理: **WP Get header**, **WP Get footer**, **WP Get body**
- カーソルの位置を指定(**WP SET FRAME**)および位置を取得(**WP Get frame**)するコマンド
- タブの先頭文字(**WP SET ATTRIBUTES** コマンドで使用)

Convert 4D View documents to 4D View Pro

Conversion

4D v16 R6 introduced the first step toward converting your 4D View documents to 4D View Pro *pre-release*. Thanks to the new **VP Convert from 4D View** command, most of the properties and information stored in 4D View documents are automatically converted, including the document structure, values, formats, styles, borders, and formulas.

These points are detailed in the following pages of the 4D View Pro Reference manual:

- **Converting 4D View documents, Conversion process**
- **Conversion details.**

4D View Pro area

A new 4D View Pro form object (**4D View Pro object**), and new associated commands are available:

- Create a new document with **VP NEW DOCUMENT**,
- Save it to disk with **VP EXPORT DOCUMENT** or in the database with **VP Export to object**,
- Then re-open it with **VP IMPORT DOCUMENT** or **VP IMPORT FROM OBJECT**.

Appendix: 4D v16 Rx Release Notes

During the R-release cycle, various libraries and components used by 4D are updated and known issues are recorded. This information, initially published via the "Release Notes" of each version, is summarized on this page.

4D v16 R2

4D Developer and 4D Volume Desktop for Windows 64-bit are in final version as of 4D v16 R2.

The 4D Pack plug-in is no longer offered in installers. Note that 4D Pack for 4D v16.x is compatible with 4D v16 R.x

4D v16 R3

The old Altura library has been removed from 4D Developer Edition and 4D Volume Desktop for Windows 64-bit.

XML: Changing the resolution of external entities, new option for **XML SET OPTIONS** (cf. **XML Modifications**).

4D v16 R4

Xerces: Update to version 3.1.4 ("Open Source" library used internally to manage XML).

The old Altura library has been removed from 4D Server for Windows 64-bit. From now on, the entire 64 bits product line is "Altura-free".

XML: As of 4D v16 R4, the **DOM Get XML element** and **DOM Count XML elements** commands are case-sensitive by default. The new XML DOM case sensitivity selector for the **XML SET OPTIONS** command allows returning to the prior behavior. (cf. **XML Modifications**).

4D v16 R5

CEF: Update to version 301. Chromium Embedded Framework (CEF).

libldap: Update to version 2.45.

libsasl: Update to version 2.1.17. (Simple Authentication Security Layer)

libzip: Update to version 1.2.

zlib: Update to version 1.2.11.(data compression software library).

HDPI screen for Windows 10 and old plug-ins:

- The old 4D Write and 4D View plug-ins are not certified for High DPI screens for Windows 10 (1709). We recommend using the 64-bit product line in this context (4D Write Pro and 4D View Pro).

Web area in 4D 32-bit and macOS 10.13:

- The web area may no longer be in focus when changing from background to the foreground. We recommend using the 64-bit product line in this context.

Rendering PDFs in a web area using 4D 32-bit for macOS 10.13:

- The web area may fail to display a PDF using the PDF viewer plug-in for macOS 10.13. We recommend using the 64-bit product line in this context.

4D v16 R6

Hunspell: Update to version 1.6.2.

The wk image, wk list style image and wk background image attributes used with the **WP GET ATTRIBUTES** or **OB Get** commands return the image itself and not its URL when the image variable is not declared. To return the image URL, use the new wk image url, wk list style image url and wk background image url attributes.

WP Get paragraphs has been renamed to **WP Create paragraph range**.

WP Get pictures has been renamed to **WP Create picture range**.

4D v17

Appendix: Useful methods for conversion

To generate a disk file listing non-indexed unique fields

[TechTip](#) to generate a disk file listing non-indexed unique fields:

```
C_LONGINT($maxTableName_l;$currentTable_l)
C_LONGINT($maxFieldCount_l;$currentField_l)
C_LONGINT($dontCare_l) // For GET FIELD PROPERTIES values that are not used.
C_BOOLEAN($dontCare_f;$isIndexed_f;$isUnique_f)
C_TEXT($logHeader_t;$logRecord_t;$logfile_t)
C_TEXT($delim_t;$lf_t)
C_TIME($logfile_h)
C_TEXT($tableName_t;$fieldName_t;$note_t)

$delim_t:=Char(Tab)
$lf_t:=Char(Carriage return)+Char(Line feed)

$logHeader_t:="Unique fields without index:"+$lf_t

$logfile_t:=Get 4D folder(Logs folder)+"UniqueNotIndexed.txt"

$logfile_h:=Create document($logfile_t)

If(OK=1)

    SEND PACKET($logfile_h;$logHeader_t)

    $maxTableName_l:=Get last table number

    For($currentTable_l;1;$maxTableName_l)
        If(Is table number valid($currentTable_l))
            $maxFieldCount_l:=Get last field number(Table($currentTable_l))
            For($currentField_l;1;$maxFieldCount_l)
                If(Is field number valid($currentTable_l;$currentField_l))

// Note the following line breaks over two lines in text,
// it is one statement in the method:
                GET FIELD PROPERTIES($currentTable_l;$currentField_l;$dontCare_l;\
```

```

$dontCare_1;$isIndexed_f;$isUnique_f;$dontCare_f)

If(($isUnique_f) & (Not($isIndexed_f)))

    $tableName_t:=Table name(Table($currentTable_1))
    $fieldName_t:=Field name(Field($currentTable_1;$currentField_1))

    $logRecord_t:="[ "+$tableName_t+" ]"+$fieldName_t+$If_t

    SEND PACKET($logfile_h;$logRecord_t)

        End if
    End if
End for
End if
End for

CLOSE DOCUMENT($logfile_h)
SHOW ON DISK($logfile_t)
End if

```

To create missing indexes

[TechTip](#) to create missing indexes on fields declared "Unique" but not indexed:

```

C_LONGINT($maxTableName_1;$currentTable_1)
C_LONGINT($maxFieldCount_1;$currentField_1)
C_LONGINT($dontCare_1) // For GET FIELD PROPERTIES values that are not used.
C_BOOLEAN($dontCare_f;$isIndexed_f;$isUnique_f)
C_TEXT($logHeader_t;$logRecord_t;$logfile_t)
C_TEXT($delim_t;$If_t)
C_TIME($logfile_h)
C_TEXT($tableName_t;$fieldName_t;$note_t)

```

```

$maxTableName_1:=Get last table number

```

```

For($currentTable_1;1;$maxTableName_1)
    If(Is table number valid($currentTable_1))
        $maxFieldCount_1:=Get last field number(Table($currentTable_1))
        For($currentField_1;1;$maxFieldCount_1)
            If(Is field number valid($currentTable_1;$currentField_1))

```

```

// Note the following line breaks over two lines in text,
// it is one statement in the method:

```

```
GET FIELD PROPERTIES($currentTable_!;$currentField_!;$dontCare_!\
$dontCare_!;$isIndexed_f;$isUnique_f;$dontCare_f)
```

```
If(($isUnique_f) & (Not($isIndexed_f)))
```

```
$tablePtr:=Table($currentTable_!)
```

```
$fieldPtr:=Field($currentTable_!;$currentField_!)
```

```
$tableName_t:=Table name($tablePtr)
```

```
$fieldName_t:=Field name($fieldPtr)
```

```
$indexName_t:="[!+$tableName_t+!]"+"$fieldName_t+" indexed for
```

```
uniqueness (kb#77023)"
```

```
ARRAY POINTER($fieldsArray_p;1)
```

```
$fieldsArray_p{1}:=!fieldPtr
```

```
CREATE INDEX($tablePtr->,$fieldsArray_p;Standard BTree
```

```
Index;$indexName_t;*)
```

```
End if
```

```
End if
```

```
End for
```

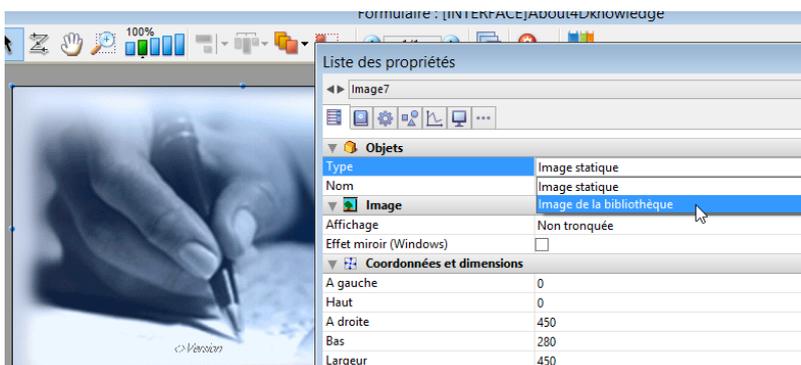
```
End if
```

```
End for
```

Conversion of picture type

To be done in 32-bit version (*i.e.*, before moving to 64 bits).

1 - Transfer your static form images to the image library:



2 - Once your images are transferred, convert them to .png or .jpeg :

```
C_LONGINT($i;$SOA;$RIS;$PictRef)
```

```
C_TEXT($PictName)
```

```
C_PICTURE($Pict)
```

```
//----- initialize arrays -----
```

```
ARRAY LONGINT($aL_PictRef;0)
```

```
ARRAY TEXT($aT_PictName;0)
```

```
ARRAY TEXT($aT_Codecs;0)
```

```
PICTURE LIBRARY LIST($aL_PictRef;$aT_PictName)
```

```
$SOA:=Size of array($aL_PictRef)
```

```

//----- convert PICT to png -----
If($SOA>0)
  For($i;1;$SOA) // for each image
    $PictRef:=$aL_PictRef{$i}
    $PictName:=$aT_PictName{$i}
    GET PICTURE FROM LIBRARY($aL_PictRef{$i};$Pict)
    GET PICTURE FORMATS($Pict;$at_Codecs)
    For($j;1;Size of array($at_Codecs))
      If($at_Codecs{$j}=".pict") // if the format is obsolete
        CONVERT PICTURE($Pict;".png") // conversion to png
    // and store in library
      SET PICTURE TO LIBRARY($Pict;$PictRef;$PictName)
    End if
  End for
End for
Else
  ALERT("The image library is empty.")
End if
//----- end of method -----

```

Merged Applications: Using Regional Settings

See [Language for commands and constants](#). In the context of deploying merged applications, to use the regional settings you must edit the contents of 4D v1x preferences file on each local machine and set the "**use_localized_language**" key to "**true**". For example:

```

$UserPreference:=Get 4D folder(Active 4D Folder)+"4D Preferences v17.4DPreferences"
$ref:=DOM Parse XML source($UserPreference;True)
$refElem:=DOM Find XML
element($ref;"preferences/com.4d/method_editor/options";arrElementRefs)&NBSP; //
Read the current value
DOM GET XML ATTRIBUTE BY NAME($refElem;"use_localized_language";$attrValue)
if($attrValue="false") // Return to <v15 behavior
  DOM SET XML ATTRIBUTE($refElem;"use_localized_language";"true")
End if
DOM EXPORT TO FILE($ref;$UserPreference)
DOM CLOSE XML($ref)

```