

廃止予定の機能と削除された機能

Rev. June 2015

このマニュアルについて

30年以上に渡り、私たちの主な目的は、新しいコンセプトやテクノロジーを用いて4Dという製品を改善することであるとともに、4Dアプリケーションの互換性を保つことにもありました。4Dを初期のころから使用している方ならご存知のことかもしれませんが、私たちは互換性に注力を注いできた結果、何年も前に古いバージョンの4DとOSで作成されたアプリケーションが最新版の製品で現在も稼働中であるのを世界中で多々見かける事ができます。

しかしながら、時として古いテクノロジーと新しいものを混ぜるのがとても難しいときもあります：


- 4Dはデベロッパに新しいテクノロジー、新しいAPI、そして新しいパラダイムを届けなければなりません。
- OSは日々変わって行きます。そして時として自らの古いAPIを廃止することもあります。


このため、4Dはときとして一部のコマンドや機能に廃止予定というタグをつける必要があります。これはいつの日か、将来のメジャーバージョンにおいてランゲージから削除されるという意味です。


デベロッパに、何が廃止予定になりその代替としてこういったものが使用できるのか知らせることによって、各々のコードを簡単に変更することができます。デベロッパには急ぐ必要もプレッシャーもなく、必要な変更を行うのに十分な時間があるからです。

ここでは現在廃止予定となっている、あるいは4D v15以降削除された機能を紹介するところから始め、その後廃止予定の機能の一覧とそれぞれについての現状を紹介します。また、廃止予定のコマンドの早見表も提供します。

また、以前の4Dのメジャーバージョンに対して作成された同様のドキュメントにアクセスすることもできます。

 v14プロダクトレンジで廃止予定の機能

 早見表

 過去のドキュメント

v14プロダクトレンジで廃止予定の機能

XSLT コマンドが廃止予定に

(4D v14 R4 以降)

関数型プログラミングコンセプトに影響されたXSLTランゲージは、XMLデータをあらゆるフォーマット(XML、HTML、または他のどんなタイプ)へと変換します。全ての主要なWebブラウザと4Dソフトウェアは、XSLT 1.0使用を実装しています。

現在XSLTは使い方もデバッグも難しいとデベロッパが感じるために、徐々に使用されない傾向にあります。このトレンドと、デベロッパからフィードバックを加味し、4Dの64bit版ではXSL変換機能は開発されないことになりました。しかしながら、4DでXSLTを引き続き使用しているお客様をサポートするために、PHP XSLライブラリーを使用する代替方法を用意しました。これは包括的なAPIを提供し、XSL変換に必要なすべてオペレーションを実行可能にするものです。このライブラリーは効率的なツールで、**`_o_XSLT APPLY TRANSFORMATION`**、**`_o_XSLT SET PARAMETER`** と **`_o_XSLT GET ERROR`** が削除されたあと、簡単に置き換えることが可能です。4Dでは、4D XSLT コマンドの代わりにPHP XSLを代用するための詳細をまとめたドキュメントをご用意しています: [Download XSLT with PHP technical document](#) (PDF).

また、HTMLページの動的な生成を扱う際には4Dタグの使用を検討することが推奨されます。何故なら、HTMLコードをフォーマットされていないテキストとして扱っている場合には、多くの場合こちらの方が簡単だからです(**MissingRef** コマンドも参照して下さい)。

注: GoogleにおけるXSLT 検索のトレンド: <https://www.google.com/trends/explore?date=all&q=xslt>

互換性のために、XSL変換は4Dにおいてまだサポートはされていますが、これからの使用は推奨されません。XSLT処理のサポートは将来の4Dのリリースにおいて削除されます。

OS X用の4D Server 64-bit版に関する注意: XSLTはOS X用の4D Server 64-bit版ではご利用になれません。結果として、XSLT コマンドのどれかをこのアプリケーションで使用した場合にはエラー33「未実装のコマンドまたはファンクションです」が生成されます。

PICT フォーマットのピクチャー

PICT フォーマットは4Dの次のメジャーリリースでサポートされなくなる予定で、4D v14では今後使用するべきではありません。その代りとして4D Pack の**AP Is Picture Deprecated**機能が、アプリケーションの移行を補助するためにv13.2以降追加されています。

注: Macの"PICT"フォーマットは、既にAppleによって数世代前のMac OSのバージョンから

廃止予定になっています(詳細に関しては、WikipediaのPICTフォーマットの項をご覧ください)。

‘PICT’フォーマットはMacのとても古いフォーマットです。バージョン11より前の4Dでは、全てのピクチャーを(Windows上でも)このフォーマットで保存してきました。PICT フォーマットは2005年にQuickDrawが廃止予定になったのに伴い廃止予定になっていました。

PICTを理解するためには一つ重要な点があります。2種類の情報を保存(または"格納"と言ってもいいでしょう)できるということです:

- 描画の基本要素そのもの(ビットマップまたはベクター)、または
- QuickTimeを使用してPICTに保存したより現代的なフォーマット(例えばJPEGなど。通常、デベロッパは **_o_QT COMPRESS PICTURE** コマンドに QT Photo compressor 定数を用いて使用していました)。

これはつまり、以前、全てのデータファイルに保存されていたピクチャーがPICTで保存されていた時代から、そのPICTには実はJPEG(または他のフォーマット)を保存することができたということです。4Dのカスタマーにとって、PICTの使用を中止することはとても大事なことです。それはPICTが既に廃止予定であるというだけではなく、PICTをWindowsで読み込むためには4DはAltura(+ **_o_QT COMPRESS PICTURE** が使用されている場合にはQuick Time)も必要になるからです。これは非効率的で、またQuickTimeのインストールも必要になります。

v11より前のバージョンのデータを変換する際には、デベロッパはデータ内の全てのピクチャーフィールドに対して **CONVERT PICTURE** コマンドを使用する必要があります。さらに、もしピクチャーフィールドが4D Write/4D Viewエリアを格納していた場合には、**CONVERT PICTURE** を特定のコーデックで使用する必要があります(コマンドのドキュメントを参照して下さい)。データを、それより新しいバージョンから変換する際には、 **AP Is Picture Deprecated** ファンクションを使用して変換が必要なピクチャーを検索することが推奨されます。

QuickTime

QuickTimeに関連したピクチャーコーデックのサポートは廃止になりました。

デフォルトでは、QuickTimeの使用は4D v14では無効化されています。しかしながら互換性のために、SET DATABASE PARAMETER、Get database parameter コマンドの新しいQuick Timeサポートオプションを使用することによって有効化する事ができます。

Windows環境下のQuickTime画像フォーマット

ここ数年、WindowsでのQuickTimeの画像管理はあまり進化していません(進化がみられるのは動画部分のみです)。4Dでは、次のリリースにおいて特にこれらのAPIのサポートを打ち切る予定です。

Windows版の4Dでは、もともと全ての主要な画像フォーマット(JPEG、PNG、GIF、TIFF、等)に加え、WIC(Windows Imaging Component)をサポートしてきました。もしあなたの

データの中に、QuickTimeにしか対応していない特定のフォーマットでWindows環境下において保存されたピクチャーがあった場合、変換することが可能です(**CONVERT PICTURE**)。

またQuickTimeピクチャーフォーマットは4D v12以降のWindows版4D Serverの64-bit版においてサポートがされていないことに注意して下さい。

Web変数に値を自動的に代入する

以前のバージョンの4Dでは、WebフォームまたはURLを通して送られた変数の値が、4D内の変数と同じ名前であったときに、Webサーバーは自動的にその値を4D変数へと代入していました。

最適化と管理の目的から、この原理は4D v14以降は維持されていません。Web変数の値は自動的に4D変数に代入されません。POST や GET を使用して送られた変数を取り戻すためには、WEB GET VARIABLES コマンドを必ず使用して下さい。送られたファイルを取り戻すためには、WEB GET BODY PART/WEB Get body part count コマンドを使用しなければなりません。

注: 自動的な代入は、v13.4以降で作成された4Dデータベースにおいてもデフォルトで無効化されています。

しかしながら、互換性のために、この機構はv13.4より前のバージョンの4Dで作成されたデータベースにおいてはデフォルトで維持されています。この場合、データベース設定の互換性ページの「Web変数に値を自動的に代入する」のオプションを使用して無効化する事ができます。

この機構は廃止されていることから、今後の変換を容易にするためにも、変換されたデータベースにおいてこのオプションのチェックを外すこと(またそれに加えて必要であればコードを修正すること)が強く推奨されます。

フォント番号

フォントを指定するのにQuickDraw ID番号を用いるのは、廃止となり、今後使用すべきではありません。Font name と Font number コマンドは互換性のために4D v14にも残されていますが、今後のバージョンで削除されます。OBJECT SET FONT コマンドはフォント名のみを受け付けます。

アプリケーションモードでのツールバー

アプリケーションモードへと切り替えた時に出演していた自動ツールバーが除去され、データベース設定のインターフェースページにあった"ツールバーを表示"オプションは削除されました。

4D v14 R5以降、カスタムのツールバーは**Open form window** コマンドに新しい`Toolbar form window` 定数を使用することで容易に使用できます。このコンテキストにおいて

は**SHOW TOOL BAR**、**HIDE TOOL BAR** そして **Tool bar height** コマンドも使用することができます。

Altura Mac2Win

Altura Mac2Win は4DをWindowsに移す際に使用されてきました。これはAPI(ファイルシステム、QuickDraw、リソース、PICTなど)を翻訳することによって、(OS X以前の)Mac OSのコードをWindowsへと移行するのを補助するためのAPIのセットです。これはかつてはとても有効で、例えばMacのプラグインデベロッパが開発したプラグインを容易にWindowsへと移行したりもできたのですが、古い(あるいは廃止された)Mac OS APIを翻訳してしまい、モダンなネイティブWindows APIを使用しません。4Dは、そのコードからMac2Winを極力除去しなければなりません。これは長く険しい道のりであり、4Dのそれぞれのバージョンにおいて、少しずつ依存性が除去されていく見込みです(そして少しずつモダンなAPIによって置き換えられていきます)。

現時点では、4Dは依然として一部をこれに依存しています(主に古いデータベースとの互換性を維持するためです)。リソース、PICT、ユーザーイベント管理の一部、Alturaを使用してビルドされたサードパーティープラグインのサポート等のために必要です。

"Resources"フォルダを内のファイルを分けるために.RSRファイル内のリソースを削除し、**CONVERT PICTURE**をPICT以外のフォーマットに変換することで、4Dデベロッパにとって4DからAlturaが除去される準備は完了します。しかしこの大変革が最も関係するのはプラグインのデベロッパです。彼らは可及的速やかにAlturaの使用をやめなければならず、これはつまり彼らのWindows用のソースコードの一部を書き換えなければいけないということです(4Dでは数年前からこれについての警告してきました)。

サブテーブル

これまでの複数のメジャーバージョンにおいて、4Dはデベロッパに対し、サブテーブルの使用をやめるよう忠告をしてきました。そして4D v11以降、サブテーブル型のフィールドを作成することは出来なくなりました。サブレコードにはいくつかの制約が付きまします。例えば、サブテーブルは必ずメモリーにロードされ、**SEND RECORD** と **DUPLICATE RECORD** コマンドで管理することはできません。

今すぐサブテーブルのサポートを打ち切る予定はありませんが、デベロッパの方々にとって、そろそろ真剣にサブテーブルをN対1テーブルへと切り替えるべき時期だと考えられます。何故なら、将来の4Dのメジャーバージョンにおいて4Dからサブテーブルが除去される予定だからです。パフォーマンスのため(リレートしたレコードの読み出しが遅いなどの特定の条件下)にサブテーブルを使用していたデベロッパは、ご安心ください。スピードは(特にv12以降)N対1、1対Nリレーションを使った方が速いです。

サブテーブルを除去するためには、主に二つのステップがあります(注:以下は完全なTech Tipsではなく、あくまで早見表です):

- v11以前のストラクチャーを変換する前に: 2004では、適当なNテーブルを作成し、1

テーブルにIDフィールドを作成します(作成されていなかった場合)。その後変える必要のあるコードを全て書き換えます(以下を参照して下さい)。

- 変換後: ここでは、4Dはサブテーブルを特殊なリレーションを使用してNテーブルに置き換え、ランゲージがサブセクションとサブレコードに作用するようになっていきます。4Dデベロッパはこの特殊なリレーションを削除し、通常のリレーションと置き換えた上で必要なコードを全て書き換えます(以下を参照して下さい)。

ここで言う"必要なコードを全て書き換える"というのは、基本的に:

- 新しいフォームを作成し、含まれるフォームをアップデート
- メソッド(プロジェクトメソッド、フォームメソッド、オブジェクトメソッド、等)において:
 - "サブレコード"テーマのコマンドを全て除去し、対応するセクションまたはレコードコマンドで置き換えます(例えば、**_o_CREATE SUBRECORD** を **CREATE RECORD** コマンドで置き換え、IDフィールドを埋めます)
 - 必要に応じてNレコードを明示的に読み出します。

注: 4D v14 R3以降、サブテーブルを含むデータベースを変換した際に4Dによって自動的に追加される特殊なフィールド"id_added_by_converter"に値を割り当てることが出来るようになります。これにより、"サブテーブルリレーション"のリンクを保ったまま、**_o_CREATE SUBRECORD**などの廃止予定のコマンドを使用することなく、リレートされたレコードを追加または編集することができます。メソッドをアップデートした後、これらの特殊なリレーションは、コードを何も書き換えることなく、通常のリレーションと置き換えることができるようになります。

非Unicodeモード

ASCII モード(この場合"非Unicodeモード"と同義)をサポートしている限り、テキストを操作する際のパフォーマンスにはあまり期待はできません。なぜなら、変換された古いストラクチャーで使用する場合は毎回 Mac-Roman へと(または Mac-Roman から)変換されなければならないからです。4Dでは将来のメジャーバージョンにおいてASCIIモードを除去する予定です。

ASCII モードは、Windows用4D Server 64-bit 版では既に除去されている点に注意して下さい。

4Dデベロッパは、変換された古いストラクチャーにおいて、Unicode モードを起動して下さい。この点に関するヒントは、[Conversion to 4D v14](#) PDFドキュメント内にあります。

また、ASCII モードは、OS X用4D Server 64-bit 版では利用できない点にも注意して下さい。

Macリソース

これもまた古いMac OSのテクノロジーの一つで、Mac OS X 10.4(Tiger、2005年)に廃止

されています。リソースは現在はテキストや文字列などの構造化されたデータ(ローカライズ用)に加え、アイコンなどを保存するために使用されています。つまり廃止されたのはリソースそのものではなくて、リソースフォークと呼ばれる、ディスク上のサポートです。リソースフォークはMac OS ファイルシステムの一部であり、Mac OS X初期から、Apple ではこのサポートを除去しようと試みてきました。これはこの仕組みがUnixやWindowsなどの他のファイルシステムとは互換性がないうえに、ファイルをネットワーク越しに転送する際に多くの問題の原因となってきたからです。

Windows上では、このメカニズムをエミュレートして、Macリソースは.RSR ファイル内に置かれてきました。

しかし、これらのリソースを管理するAPIがまだあるとはいえ(そしてMac OSではデータフォーク内に保存されているリソースをひそかに管理しているとはいえ)、以下の複数の理由からこの機構は使用されるべきではありません:

- テキストと文字列はMac-Romanになっています。TEXT または STR# タイプのリソースにはUnicodeを保存することができません。
- PICT リソースは PICT フォーマットを保存します。これは現代的ではなく、透明度もありません(上記の"PICT フォーマットのピクチャー"を参照して下さい)。
- リソースには数・サイズともに上限があります(約2,700リソース、または16MB)。

4Dからはリソースを書き込み・作成するコマンドは除去されています。

リソースを依然として使用している4Dアプリケーションの大半は、実際に使用しているのは"String List"リソース('STR#')です。4Dでは STR# を XLIFF に容易に変換するツールを提供しています:

- 4D Pop コンポーネントは、STR# ファイルの中身を読んで変換することで自動的に XLIFF ファイルを作成する事ができます。
- STR# を参照する全てのルーチンと式は、何の変換もなしにそのまま XLIFF を使用する事ができます。例えば、ボタンやメニューのラベルが":15000,3" (=「STR# ID 15000 の3つめの項目を取得しなさい」という意味) だった場合、4D は適切な XLIFF を(存在すれば)読み込みます。

他のリソースに関して:

- リソースをResources フォルダ内の他のファイル形式に格納します(必要であればサブディレクトリを作成して下さい):
 - 'TEXT' リソースを XLIFF または .txt ファイル形式に保存します。
 - 'PICT' リソースを別の.jpg/.png などのファイル形式に保存します。
 - 'PICT' + MASK' リソースを .png ファイルとして保存します。
 - ICON またはカラー付きアイコンの代わりに、icns を使用します(Mac)。
 - 他のプライベートリソースを、適切な形で保存します(多くの場合、特定の拡張子付きのバイナリーファイルとして保存します)。
- "Resources" フォルダにリソースを保管します。リソースの親のパスを動的に取得する場合には **Get 4D folder(Current resources folder)** コマンドを使用します。

プラグインのためのAPI QuickDraw

プラグインには二つの種類があります: 新しいプラグインAPIを使用するものと、古いAPI(と QuickDraw)を使用するものです。

古いツールボックス(とQuickDraw)を使用するプラグインに対しては、互換性を維持するため、描画/レンダリングは以前のバージョンのように QuickDraw ポート内で直接行われることはありません。その代り、プラグインに割り当て有られたGWorld QuickDraw オフスクリーンエリアを通して行われます。

その結果、プラグインはコンテナ(フォームオブジェクト)によって設定されたカレントのポートを変更してはいけな、などのいくつかのルールに従う必要があります。

新しいツールボックスを使用するプラグインに関しては、その新しいツールボックスのみが使用され、QuickDraw は使用されません

(http://sources.4d.com/trac/4d_4dpluginapi/wiki/native_drawing を参照して下さい)。

ランゲージ: 廃止予定または削除されたコマンド

廃止・削除のコマンド	代替機能	廃止時期	現在の状況
4D 環境テーマ:			
_o_ADD DATA SEGMENT	-	v11	廃止予定
_o_DATA SEGMENT LIST	-	v11	廃止予定
コンパイラーテーマ:			
_o_ARRAY STRING	ARRAY TEXT	v12	廃止予定
C_GRAPH	(GRAPH コマンドとSVGを併用して下さい)	v12	廃止予定
_o_C_INTEGER	C_LONGINT	v12	廃止予定
_o_C_STRING	C_TEXT (データベースがUnicodeモードになれば使用可能)	v12	廃止予定
データ入力テーマ:			
_o_ADD SUBRECORD	N->1 リレーションのテーブルにおいて ADD RECORD	v12	廃止予定
Modified	Form event と <u>On data change</u>	v12	廃止予定

_o_MODIFY SUBRECORD	N->1 リレーションのテーブルにおいて MODIFY RECORD	v12	廃止予定
フォームイベントテーマ:			
Activated	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
After	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
Before	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
Deactivated	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
_o_During	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
In break	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
In footer	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
In header	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
Outside call	Form event と適切なイベントの組み合わせで置き換え	v12	廃止予定
グラフテーマ:			
GRAPH (4D Graphエリアを使用)	代わりにSVGピクチャーを使用	v12	廃止予定
_o_GRAPH TABLE	データを配列にビルドし、SVGピクチャー内で GRAPH を使用	v13	4D v14では無効化
階層リストテーマ:			
_o_REDRAW LIST	コードから除去(v11以降何の動作もしていません)	v11	廃止予定
オブジェクト(フォーム)テーマ:			
_o_DISABLE BUTTON/_o_ENABLE BUTTON	OBJECT SET ENABLED	v12	廃止予定

ピクチャーテーマ:

_o_PICTURE TYPE LIST	PICTURE CODEC LIST	v12	廃止予定
_o_QT COMPRESS PICTURE	CONVERT PICTURE	v12	廃止予定
_o_QT COMPRESS PICTURE FILE	WRITE PICTURE FILE/PICTURE TO BLOB	v12	廃止予定
_o_QT LOAD COMPRESS PICTURE FROM FILE	READ PICTURE FILE/CONVERT PICTURE	v12	廃止予定
_o_SAVE PICTURE TO FILE	WRITE PICTURE FILE	v12	廃止予定

リソーステーマ: リソースを書き込み/作成する全てのコマンド:

_o_ARRAY TO STRING LIST	-	v12	廃止予定
_o_Create resource file	-	v12	廃止予定
_o_DELETE RESOURCE	-	v12	廃止予定
_o_Get component resource ID	-	v12	廃止予定
_o_SET PICTURE RESOURCE	-	v12	廃止予定
_o_SET RESOURCE	-	v12	廃止予定
_o_SET RESOURCE NAME	-	v12	廃止予定
_o_SET RESOURCE PROPERTIES	-	v12	廃止予定
_o_SET STRING RESOURCE	-	v12	廃止予定
_o_SET TEXT RESOURCE	-	v12	廃止予定

SQL テーマ:

_o_USE EXTERNAL DATABASE	SQL LOGIN	v12	廃止予定
---------------------------------	------------------	-----	------

_o_USE INTERNAL DATABASE	SQL LOGOUT	v12	廃止予定
文字列テーマ:			
_o_Convert case	CONVERT FROM TEXT/Convert to text を必要に応じて使用して下さい	v11	廃止予定
_o_ISO to Mac	変換が必要ない場合はメソッドからコマンドを除去して下さい	v11	廃止予定
_o_Mac to ISO	(これはつまりデータベースをUnicodeモードで実行するという意味です)	v11	廃止予定
_o_Mac to Win		v11	廃止予定
_o_Win to Mac		v11	廃止予定
サブレコードテーマ: 全コマンド	“nnn SUBRECORD” と “nnnSUBSELECTION” を、N->1リレーションにあるNテーブルのNレコードまたはNテーブルのアクションで置き換え	v12	廃止予定
システムドキュメントテーマ:			
Document type	-	v12	廃止予定
システム環境テーマ:			
_o_Font name	フォント識別子を使用	v14	廃止予定
_o_Font number	QuickDraw の廃止に伴い、 _o_Font name と _o_Font number コマンドも廃止になります OBJECT SET FONT コマンドは、フォントに対し倍長整数の引数を受け取らなくなりました。この引数の型は文字列になり、フォント名を指定する必要があります。	v14	廃止予定
ユーザーインターフェーステーマ:			
_o_Get platform interface/_o_SET PLATFORM INTERFACE	変換されたアプリケーション内でのみ使用可能(<u>Automatic Platform</u> 定数と併用して下さい)	v12	廃止予定

HIDE TOOL BAR/SHOW TOOL BAR	-	-	v14では無効化
Web Serverテーマ:			
_o_SET CGI EXECUTABLE	-	v12	v13では無効化
_o_SET WEB DISPLAY LIMITS	非コンテキストモードを使用して下さい	v12	v13では無効化
_o_SET WEB TIMEOUT	非コンテキストモードを使用して下さい	v12	v13では無効化
_o_Web Context	非コンテキストモードを使用して下さい	v12	v13では無効化
XML テーマ:			
_o_XSLT APPLY TRANSFORMATION	PHP <i>libxslt</i> モジュールまたはMissingRef コマンドを使用して下さい	v14R4	v14R4では廃止予定
_o_XSLT GET ERROR	PHP <i>libxslt</i> モジュールまたはMissingRef コマンドを使用して下さい	v14R4	v14R4では廃止予定
_o_XSLT SET PARAMETER	PHP <i>libxslt</i> モジュールまたはMissingRef コマンドを使用して下さい	v14R4	v14R4では廃止予定
4D PACK コマンド:	(詳細に関しては、 4D Pack から削除された コマンド を参照して下さい)		
AP AVAILABLE MEMORY	GET MEMORY STATISTICS	-	v14では削除されています
AP HELP INDEX	Windows Vistaと互換性のないコマンドです。このコマンドの使用は推奨されません。	-	v14では削除されています
	Windows Vistaと互換性のないコマンドで		v14で

AP HELP ON HELP	す。このコマンドの使用は推奨されません。	-	は削除されています
AP HELP ON KEY	Windows Vistaと互換性のないコマンドです。このコマンドの使用は推奨されません。	-	v14では削除されています
AP CLOSE HELP	Windows Vistaと互換性のないコマンドです。このコマンドの使用は推奨されません。	-	v14では削除されています
AP Create method	METHOD SET CODE と METHOD SET ATTRIBUTE	-	v14では削除されています
AP Modify method		-	v14では削除されています
AP Does method exist	METHOD GET NAMES	-	v14では削除されています
AP Get picture type	" ピクチャ " テーマのコマンド	-	v14では削除されています
AP Get templates	廃止	-	v14では削除されています
_AP External clock	タイムピッカーウィジェット	-	v14では削除されています

AP SET CLOCK	タイムピッカーウィジェット	-	は削除されています
AP Rect Dragger	SET DRAG ICON	-	v14では削除されています
AP Timestamp to GMT	<pre>\$myDate:=String(Current date;ISO Date GMT;Current time) // 返り値は、例えば以下のようになります "2014-05-06T12:19:23Z"</pre>	-	v14では削除されています
AP SET PARAM	廃止	-	v14では削除されています
AP GET PARAM	廃止	-	v14では削除されています

Obsolete commands renamed and hidden

廃止予定のコマンドとは、使用の中止が強く推奨されるようになり、将来のバージョンの4Dにおいては維持されない予定のものです。

4D v15では分かりやすさのために、今までついていなかったものも含めて全ての廃止予定のコマンドに"_o_"の接頭辞をつけました。

廃止予定のコマンドは4Dリストには表示されなくなったため(廃止予定のコマンドは非表示にの章を参照して下さい)、それらを選択することはできなくなりました。廃止予定のコマンドが既存のコードの中で使用されていた場合、それらは単純に改名されます。

以下は、4D v15で改名された廃止予定のコマンドの一覧です(ここにあるものは4Dリストには表示されません):

以前の名前	4D v15での新しい名前
ADD DATA SEGMENT	_o_ADD DATA SEGMENT
ADD SUBRECORD	_o_ADD SUBRECORD
ALL SUBRECORDS	_o_ALL SUBRECORDS
APPLY TO SUBSELECTION	_o_APPLY TO SUBSELECTION
ARRAY STRING	_o_ARRAY STRING

ARRAY TO STRING LIST	_o_ARRAY TO STRING LIST
Before subselection	_o_Before subselection
C_INTEGER	_o_C_INTEGER
C_STRING	_o_C_STRING
Convert cas	_o_Convert case
Create resource file	_o_Create resource file
CREATE SUBRECORD	_o_CREATE SUBRECORD
DATA SEGMENT LIST	_o_DATA SEGMENT LIST
DELETE RESOURCE	_o_DELETE RESOURCE
DELETE SUBRECORD	_o_DELETE SUBRECORD
DISABLE BUTTON	_o_DISABLE BUTTON
During	_o_During
ENABLE BUTTON	_o_ENABLE BUTTON
End subselection	_o_End subselection
FIRST SUBRECORD	_o_FIRST SUBRECORD
Font name	_o_Font name
Font number	_o_Font number
Get component resource ID	_o_Get component resource ID
Get platform interface	_o_Get platform interface
INVERT BACKGROUND	_o_INVERT BACKGROUND
ISO to Mac	_o_ISO to Mac
LAST SUBRECORD	_o_LAST SUBRECORD
Mac to ISO	_o_Mac to ISO
Mac to Win	_o_Mac to Win
MODIFY SUBRECORD	_o_MODIFY SUBRECORD
NEXT SUBRECORD	_o_NEXT SUBRECORD
ORDER SUBRECORDS BY	_o_ORDER SUBRECORDS BY
PICTURE TYPE LIST	_o_PICTURE TYPE LIST
PREVIOUS SUBRECORD	_o_PREVIOUS SUBRECORD
QT COMPRESS PICTURE	_o_QT COMPRESS PICTURE
QT COMPRESS PICTURE FILE	_o_QT COMPRESS PICTURE FILE
QT LOAD COMPRESS PICTURE FROM FILE	_o_QT LOAD COMPRESS PICTURE FROM FILE
QUERY SUBRECORDS	_o_QUERY SUBRECORDS
Records in subselection	_o_Records in subselection
REDRAW LIST	_o_REDRAW LIST

SAVE PICTURE TO FILE	_o_SAVE PICTURE TO FILE
SET PICTURE RESOURCE	_o_SET PICTURE RESOURCE
SET PLATFORM INTERFACE	_o_SET PLATFORM INTERFACE
SET RESOURCE	_o_SET RESOURCE
SET RESOURCE NAME	_o_SET RESOURCE NAME
SET RESOURCE PROPERTIES	_o_SET RESOURCE PROPERTIES
SET STRING RESOURCE	_o_SET STRING RESOURCE
SET TEXT RESOURCE	_o_SET TEXT RESOURCE
USE EXTERNAL DATABASE	_o_USE EXTERNAL DATABASE
USE INTERNAL DATABASE	_o_USE INTERNAL DATABASE
Win to Mac	_o_Win to Mac

_o_XSLT APPLY TRANSFORMATION

```
_o_XSLT APPLY TRANSFORMATION ( xmlSource ; xslSheet ; result {; compileSheet}  
)
```

引数	型	説明
xmlSource	文字, BLOB	➡ XMLソースドキュメントの名前またはアクセスパス、またはXMLソースを含むBLOB
xslSheet	文字, BLOB	➡ XSLスタイルシートドキュメントの名前または アクセスパス、または XSLスタイルシートを含むBLOB
result	文字, BLOB	➡ XSL変換の結果を受け取るドキュメントの名前または アクセスパス、または XSL変換の結果を受け取るBLOB
compileSheet	ブール	➡ True = XSL変換を最適化 Falseまたは省略 = 最適化しない、 コンパイルされたXSLファイルがあれば削除する

互換性に関する注意

4D v14 R4 以降、XSL 変換コマンドは廃止予定になります。互換性のためにサポートはされますが、速やかに使用を中止することが推奨されます。近い将来の4DのバージョンにおいてXSLTテクノロジーは使用不可になります。詳細な情報に関しては、[XMLユーティリティコマンドの概要](#)を参照して下さい。

⚙️ AP Is Picture Deprecated

AP Is Picture Deprecated (ptrPict) -> 戻り値

引数	型	説明
ptrPict	ポインター	➡ テストするピクチャーフィールドや変数へのポインター
戻り値	倍長整数	🔄 0 = 有効なフォーマット、1 = 無効なフォーマット (PICT)

説明

AP Is Picture Deprecated コマンドは *ptrPict* 引数により参照されるピクチャーが (古いフォーマットである) PICTフォーマットを含む場合1を、そうでなければ0を返します。ピクチャーがPICTフォーマットである場合、あるいはPICTフォーマットを含む場合 (ピクチャー内には複数のフォーマットが含まれる場合があります) に、古いフォーマットと判定されます。

ptrPict には、4Dピクチャーフィールドや変数、あるいはピクチャー配列の要素へのポインターを渡します。

このコマンドは、4Dの次のメジャーバージョンの準備をするために用意されました。Macの"PICT"フォーマットはAppleにより、すでに廃止予定としてアナウンスされています ([WikipediaのPICTフォーマットの記事参照](#))。互換性の目的で4DはPICTフォーマットをサポートしてきましたが、4D v14以降はサポートされません。アプリケーション移行を開始するために、このコマンドを使用してPICTピクチャーを探し、**CONVERT PICTURE**コマンドを使用して現在利用可能なピクチャーに変換することができます。変換後ピクチャーには指定したフォーマットのみが含まれます。

注: ピクチャーが印刷に使用されていた場合、PICTフォーマットを削除する前に事前の検証が必要です。PICTはベクターフォーマットであったため、印刷目的の場合にはEMFなどの他のベクターフォーマットに変換することが推奨されます。

例題

ライブラリ中のピクチャーをテストし、必要であれば変換を行う:

```
C_LONGINT($v1Pict;$vPictCount;$vPictRef;$vTotal)
C_TEXT($vPictName)
C_POINTER($vPointer)
C_PICTURE($vpPict)
ARRAY TEXT($arrPictNames;0)
ARRAY LONGINT($arrPictRefs;0)

$vTotal:=0
```

```

PICTURE LIBRARY LIST($arrPictRefs;$arrPictNames)
$vPictCount:=Size of array($arrPictRefs)
If($vPictCount>0)
    For($vIPIct;1;$vPictCount) // 各ピクチャーごとに
        $vPictRef:=$arrPictRefs{$vIPIct}
        $vPictName:=$arrPictNames{$vIPIct}
        GET PICTURE FROM LIBRARY($arrPictRefs{$vIPIct};$vpPict)
        $vPointer:=->$vpPict // ポインターを渡す
        $isObs:=AP Is Picture Deprecated($vPointer)
        If($isObs=1) // フォーマットが廃止されていれば
            CONVERT PICTURE($vPointer->".PNG") // png へと変換
        // そしてライブラリに保存
            SET PICTURE TO LIBRARY($vPointer->,$vPictRef;$vPictName)
            $vTotal:=$vTotal+1
        End if
    End for
    ALERT(String($vTotal)+" picture(s) out of "+String($vPictCount)+"
were converted to png.")
Else
    ALERT("The picture library is empty.")
End if

```

_o_QT COMPRESS PICTURE

_o_QT COMPRESS PICTURE (picture ; method ; quality)

引数	型		説明
picture	ピクチャー	⇒	圧縮するピクチャ
		⇐	圧縮されたピクチャ
method	文字	⇒	4文字の圧縮方法
quality	倍長整数	⇒	圧縮品質 (1..1000)

互換性に関する注意

このコマンドは廃止されたメカニズムを呼び出します。そのためこのコマンドは**CONVERT PICTURE** コマンドで置き換えなければなりません。

CONVERT PICTURE

CONVERT PICTURE (picture ; codec {; compression})

引数	型		説明
picture	ピクチャー	→	変換するピクチャ
		←	変換されたピクチャ
codec	文字	→	ピクチャCodec ID
compression	実数	→	圧縮の品質

説明

CONVERT PICTURE コマンドは`picture`を新しいタイプに変換します。

`codec` 引数で生成するピクチャのタイプを指定します。Codecには拡張子 (例 “.gif”), または Mimeタイプ(例 “image/jpeg”)が使用できます。利用可能なCodecのリストは **PICTURE CODEC LIST** コマンドを使用して取得できます。

`picture` フィールドや変数が複合型 (例えばコピー/ペーストアクションの結果のピクチャ) の場合、`codec`タイプに対応する情報のみが結果ピクチャに保持されます。

Note: リクエストされた `codec` のタイプが`picture`の元のタイプと同じ場合、変換は実行されず、そのままのピクチャが返されます (`compression`引数が使用された場合を除く、後述参照)。

オプションの`compression`引数が渡されると、互換性のあるCODEC利用時に、結果のピクチャに適用する圧縮品質を指定できます。`compression`には圧縮品質を指定する0から1の間の値を渡します。0が圧縮優先で1が品質優先です。この引数はCODECが圧縮をサポートし (例えばJPEGやHDPhoto)、さらにWICやImageIO APIによりサポートされている場合にのみ考慮されます。ピクチャ管理APIに関する詳細情報は、**ピクチャ** を参照してください。`compression`引数を省略するとデフォルトで品質優先 (`compression = 1`) が適用されます。

例題 1

vpPhoto ピクチャをjpegフォーマットに変換:

```
CONVERT PICTURE (vpPhoto; ".jpg")
```

例題 2

60%の品質でピクチャを変換:

```
CONVERT PICTURE(vPicture;".JPG";0.6)
```


Open form window

Open form window ({aTable ;} formName {; type {; hPos {; vPos {; *}}}}) -> 戻り値

引数	型	説明
aTable	テーブル	→ フォームが属するテーブル、または省略時デフォルトテーブル
formName	文字	→ フォーム名
type	倍長整数	→ ウィンドウタイプ
hPos	倍長整数	→ ウィンドウの横位置
vPos	倍長整数	→ ウィンドウの縦位置
*	演算子	→ ウィンドウの現在の位置とサイズを保存
戻り値	WinRef	→ ウィンドウ参照番号

説明

Open form window コマンドはフォーム *formName* のサイズとリサイズプロパティを使用して、新しいウィンドウを開きます。

formName フォームはウィンドウに表示されません。フォームを表示するには、フォームをロードするコマンド (**ADD RECORD** 等) を呼び出さなければなりません。

デフォルトで (*type* 引数が渡されないと)、クローズボックス付きの標準ウィンドウが開かれます。 **Open window** コマンドと異なり、クローズボックスにはメソッドは割り当てられません。クローズボックスをクリックすると、On Close Box フォームイベントが有効にされている場合を除き、ウィンドウがキャンセルされ閉じられます。 On Close Box フォームイベントが有効であれば、割り当てられたコードが実行されます。

formName フォームがリサイズ可能であれば、開かれるウィンドウにはズームボックスとグローボックスが付加されます。

Note: フォームの主なプロパティを取得するには **FORM GET PROPERTIES** コマンドを使用します。

任意の *type* 引数は、ウィンドウのタイプを指定するために使用します。以下のいずれかの定数を渡さなければなりません (**Open Form Window** テーマ内):

定数	型	値
Form has full screen mode Mac	倍長整数	65536
Modal form dialog box	倍長整数	1
Movable form dialog box	倍長整数	5
Palette form window	倍長整数	1984
Plain form window	倍長整数	8
Pop up form window	倍長整数	32
Sheet form window	倍長整数	33
Toolbar form window	倍長整数	35

Notes:

- (グローボックス, クローズボックスなど...) 作成されたウィンドウの属性は、選択された *type* に対する OS のインタフェース仕様に基づきます。ゆえに使用するプラットフォームによって異なる結果が得られる場合があります。
- Has toolbar button Mac OS と Has full screen mode Mac 定数は、他の定数に加算して使用しなければなりません。
- ウィンドウタイプに関する詳細は **ウィンドウタイプ** のセクションを参照してください。ただし **Open Form Window** テーマの定数のみが **Open form window** コマンドで利用可能であることに注意してください。

Toolbar form window 定数が渡された場合、ウィンドウの位置、サイズ、グラフィックプロパティはツールバーのそれに準拠します。具体的には、以下の用になります:

- ウィンドウは常にメニューバーのすぐ下に表示されます。
- ウィンドウの水平方向のサイズはデスクトップ(OS X)または4Dのメインウィンドウ内部(Windows)の利用可能な全ての水平方向のスペースを埋める形で自動的に調整されます。ウィンドウの垂直方向のサイズは、他の全てのフォームウィンドウタイプ同様、フォームプロパティに準じます。
- ウィンドウに境界線はなく、手動で移動・リサイズはできません。また、*hPos*、*vPos* そして * 引数は渡されていた場合でも無視されます。
- - 二つの異なるツールバーウィンドウを同時に作成することはできません。もしツールバーウィンドウが既に存在している状態で、**Open form window** に Toolbar form window type が渡された状態で呼び出された場合、エラー-10613("ツールバー型のフォームウィンドウを二つ作成することはできません")が生成されます。

OS Xフルスクリーンモードでのツールバーフォームウィンドウ: お使いのアプリケーションがツールバーウィンドウと、フルスクリーンモードをサポートする(Has full screen mode Mac オプション)標準のウィンドウの両方を表示する場合、標準のウィンドウがフルスクリーンモードに切り替わった時にはインターフェースのルールに則り、ツールバーを非表示にしなければなりません。ウィンドウがフルスクリーンモードに切り替わったかどうかを調べるためには、ウィンドウの垂直方向のサイズがスクリーンの高さと同じになったかをテストして

下さい(詳細は**HIDE TOOL BAR** コマンドを参照して下さい)。

オプションの引数 *hPos* を使用して、ウィンドウの横位置を指定できます。特定の位置をピクセル単位で指定するか (**Open window** コマンド参照)、 **Open Form Window** テーマの以下の定義済み定数を渡します:

定数	型	値
Horizontally centered	倍長整数	65536
On the left	倍長整数	131072
On the right	倍長整数	196608

オプションの引数 *vPos* を使用して、ウィンドウの縦位置を指定できます。特定の位置をピクセル単位で指定するか (**Open window** コマンド参照)、 **Open Form Window** テーマの以下の定義済み定数を渡します:

定数	型	値
At the bottom	倍長整数	393216
At the top	倍長整数	327680
Vertically centered	倍長整数	262144

これらの引数はツールバーとメニューバーの表示状態、およびWindowsではアプリケーションウィンドウ現在のサイズを考慮します。

オプションの引数 *** を渡すと、閉じられるときにその時点での位置とサイズが記憶されます。ウィンドウが再度開かれると、以前の位置とサイズが再現されます。この場合、*vPos* と *hPos* 引数はウィンドウが最初に開かれるときにのみ使用されます。

例題 1

以下のコードはクローズボックス付きの標準のウィンドウを開き、自動で"Input"フォームのサイズに調整します。フォームはリサイズ可能に設定されているので、ウィンドウはグローとズームボックスを持ちます:

```
$winRef :=Open form window([Table1];"Enter")
```

例題 2

以下のコードはプロジェクトフォーム"Tools"に基づき、スクリーンの左上の位置にフローティングパレットを開きます。このパレットは閉じられた時の位置を記憶し、再度開かれるときにはその位置が使用されます:

```
$winRef :=Open form window("Tools";Palette form window;On the left;At the top;) 
```

```
SEND RECORD {( aTable )}
```

引数	型	説明
aTable	テーブル	→ カレントレコードを送信するテーブル, または 省略した場合デフォルトテーブル

説明

SEND RECORD は、*aTable*のカレントレコードを**SET CHANNEL**コマンドで開いたシリアルポートまたはドキュメントに送信します。レコードは特別な内部フォーマットで送信され、**RECEIVE RECORD**でのみ読み込むことができます。カレントレコードが存在しなければ、**SEND RECORD**は何も行いません。

完全なレコードを送信します。つまりレコードにあるいはレコードとともに格納されたピクチャやBLOBも送信します。


重要: **SEND RECORD** と **RECEIVE RECORD**を使用してレコードが送受信される場合、送信元と送信先のテーブル構造は互換性のあるものでなくてはなりません。互換性がない場合、**RECEIVE RECORD**コマンドの実行時4Dがテーブル定義に応じて値を変換します。

Notes: このコマンドを使用してドキュメントにレコードを送信する場合、ドキュメントは**SET CHANNEL**コマンドを使用して開かれていなければなりません。**Open document**, **Append document** または **Create document**で開かれたドキュメントに対して、**SEND RECORD**を使用することはできません。

互換性に関する注意: 4Dバージョン11より、このコマンドはサブテーブルをサポートしません。

例題

RECEIVE RECORDコマンドの例題を参照

_o_CREATE SUBRECORD (subtable)		
引数	型	説明
subtable	サブテーブル	 新しいサブレコードを作成するためのサブテーブル

互換性に関するメモ

バージョン11以降の4Dはサブテーブルをサポートしていません。互換性メカニズムは、変換されたデータベースでコマンドの機能を保護しますが、すべてのサブテーブルは、リレートする標準的なテーブルに取り換えられることが強く推奨されます。

_o_ADD DATA SEGMENT

_o_ADD DATA SEGMENT

このコマンドは引数を必要としません

説明

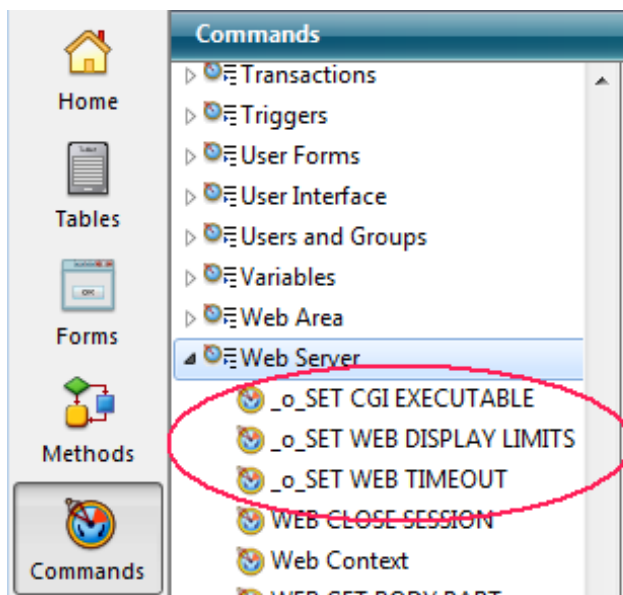
互換性に関する注記: 4Dバージョン11より、データセグメントはサポートされていません (データファイルのサイズに制限はありません)。このコマンドは呼び出されても何も行いません。

📄 廃止予定のコマンドは非表示に

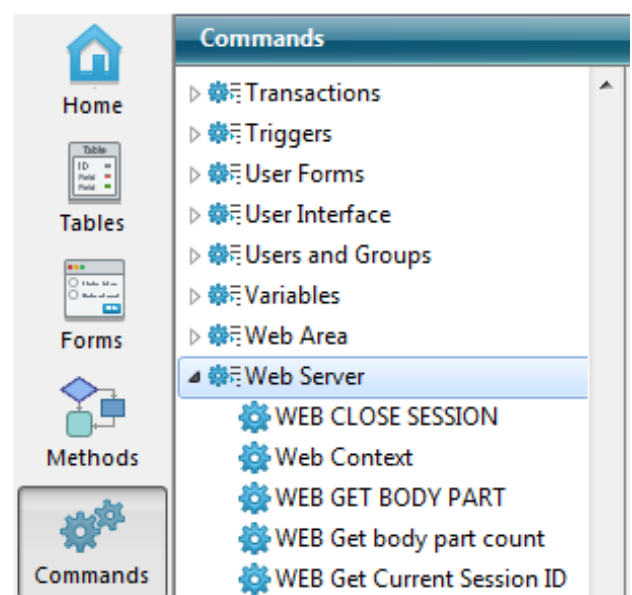
4D v15以降、全ての廃止予定(または廃止された)コマンドはシステム的にコマンド名の先頭に"_o_"がつけられ、4Dのリストとコンプリートリスト内に表示されなくなります。

注: 4D v15で改名された廃止予定のコマンドの一覧は、"**廃止予定のコマンドの改名**"のセクションにて確認することができます。

以前のリリースにおいては、廃止予定のコマンドは必ずしもコマンド名が改変されていたわけではなく、通常のコマンドと並んで一覧に表示されていたため、混乱の原因となる可能性があります。4D v15では、廃止予定になっていないコマンドのみコマンドの一覧から選択することができます:



4D v14



4D v15

関係するリストは以下の箇所です:

- エクスプローラー/"コマンド"タブ
- メソッドエディターのヘルパーリスト(コマンドの一覧と、テーマごとのコマンドの一覧)、それから入力補完ポップアップ
- フォーミュラエディターヘルパーリスト

もちろん、既存のコードの中で、廃止予定のコマンドが勝手に削除されていることはありません(ただし先頭に"_o_"がつけられています)し、サポートされている限りはドキュメントにある通りの振る舞いを続けます。必要であれば、メソッド内でその名前をタイプするだけで廃止予定のコマンドを使用することもできます。これらは引き続き正常に解釈されます:

しかしながら、廃止予定のコマンドを新しいコードの中で使用することは推奨されていないという点に注意して下さい([廃止予定の機能と削除された機能](#)を参照して下さい)。

早見表

"状況"カラムの値についての説明です:

- **削除済み**: カレントのバージョン(または記載されたバージョン)ではデフォルトで使用できなくなっています。
- **廃止予定**: 今後使用されるべきではなく、将来のメジャーバージョンにおいて削除される予定です。
- **OS**: OSにおいて廃止予定のテクノロジーの状況に拠ります(例:PICT フォーマット)。状況としては"廃止予定"と同等ですが、4Dがそれを削除する前にOSが除去することが有り得ます。

機能	代替手段	状況
アプリケーションモードの自動ツールバー	Open form window と <u>Toolbar form window</u>	4D v14 以降 削除 済み
Mac OS QuickDraw フォント	フォント名	廃止 予定
Altura Mac2Win	プラグイン開発者の方へ: ネイティブ Windowsコードを使用して下さい	廃止 予定
Web変数に値を自動的に代入する(v13.4以前に作成されたデータベース用の互換性オプション)	WEB GET VARIABLES コマンド (変数を取得) WEB GET BODY PART/WEB Get body part count コマンド(ポストしたファイルを取得)	廃止 予定
非Unicodeモード(変換されたv11以前のデータベース)	Unicodeに移行	廃止 予定
QuickTime サポート(互換性オプションは、 SET DATABASE		廃止

PARAMETER (QuickTimeSupport;1)を使用することにより利用可能)	ネイティブフォーマットの使用	予定
API QuickDraw	サードパーティプラグインには新 SDK プラグインを使用して下さい	廃止 予定
サブテーブル	N->1 テーブルの使用	廃止 予定
XSLT	PHP <i>libxslt</i> モジュールまたは PROCESS 4D TAGS コマンドを使用して下さい	廃止 予定
Mac リソース(cicn、PICT)	"Resources" フォルダの使用 互換性のために、変換されたデータベース内では使用可能です。ただし書き込みアクセスコマンドはサポートされません	OS
PICT	新しいフォーマットを使用して下さい。詳細に関しては AP Is Picture Deprecated	OS

過去のドキュメント

このドキュメントは、**4D v15 プロダクトレンジ**に関するドキュメントです。参照のために、以前のプロダクトレンジで廃止になった機能について言及している過去のドキュメント (PDF、英文) を閲覧することができます。リンクは以下の通りです:

- [Deprecated and Removed Features in 4D v14](#) - (Rev. Oct 2014)
- [Deprecated and Removed Features in 4D v13](#) - (Rev. 20 Feb 2012)
- [Deprecated and Removed Features in 4D v12](#) - (Rev. 03 Jun 2010)