

# ❏ 廃止予定の機能と削除された機能

Rev. January 2017

## 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こと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ど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 v16 R2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。

❏ v16プロダクトレンジで廃止予定または削除された機能

❏ 早見表

❏ 過去のドキュメント

NEW 新着

# 📄 v16プロダクトレンジで廃止予定または削除された機能

## 4D Developer Edition 64-bit 版について

---

4D の 64-bit版は最近のテクノロジーを多分に採用しており、以前のバージョンの4Dで廃止予定と宣言されたテクノロジーについては通常サポートしていません。64-bit版の4Dのプロダクトレンジでサポートされていない機能の完全な一覧については、デザインリファレンスマニュアルの**64-bit版のみの機能**の章を参照して下さい。

## XSLT コマンドが廃止予定に

---

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: [#command\\_5](#)

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。

**64-bit版に関する注意:** XSLTはOS X用およびWindows用の64-bit版4Dと、OS X用64-bit版4D Server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 では今後使用するべきではありません。**GET PICTURE FORMATS**コマンドを使用すると、データファイル内でPICTフォ

ーマットを使用しているピクチャーを検知しフィルターすることができます (4D Pack の **\_o\_AP Is Picture Deprecated** 機能は廃止予定となりました)。

**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)。それより新しいバージョンからデータを変換する際には、**GET PICTURE FORMATS** コマンドを使用して変換が必要なピクチャーをデータファイルより検出することが推奨されます。

## データベースストラクチャー内の PICT フォーマットを検出

v16以降では **Maintenance & Security Center (MSC)** を使って、データベースストラクチャーに含まれる、廃止された PICT フォーマットのピクチャーを検出できます。**アプリケーションの検証** 機能を使用したときに PICT フォーマットのピクチャーを検出した場合、生成されるログファイルにはその旨の警告が含まれます。これらの警告は静的なピクチャーのほかにも、ピクチャーライブラリおよびフォームオブジェクトに含まれているピクチャーが対象となります。

**注:** 廃止された PICT フォーマットが使用されているピクチャーの削除・置換は別途行う必要があります。**MSC** で**ストラクチャーファイルの修復** 処理を行っても、"廃止" ピクチャーは影響されず、ログファイルには引き続き警告が記録されます。

## 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 (64-bit 版の 4D では QuickTime がサポートされていないため、有効化できません)。

## 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。**\_o\_Font number**と**\_o\_Font name**コマンドは互換性のために4D v15以降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。

## 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います。

ASCII モードは、Windows用4D Server 64-bit 版のコンパイル済みストラクチャー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。

**64-bit版に関する注意:**また、ASCII モードは、64-bit版の4Dおよび4D Server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、透明度也没有ありません(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](http://sources.4d.com/trac/4d_4dpluginapi/wiki/native_drawing) を参照して下さい)。

## 4D Pack

---

これまでの異なるバージョンを経てきた中で、有用な4D Packルーチンは徐々に4D自身へと統合される一方、古くなった機能は削除されてきました。4D Pack v16にはごく少数のルーチンのみが含まれ、これらは今後開発が進められることはありません。4D v16以降、4D Pakプラグイン全体が廃止予定となり、今後のバージョンにおいては提供されなくなります。残りのルーチンに対して利用可能な代替ソリューション(あれば)については、以下の表を参照してください。

## 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:			
<b>_o_ADD DATA SEGMENT</b>	-	v11	廃止予定
<b>_o_DATA SEGMENT LIST</b>	-	v11	廃止予定
バックアップテーマ:			
<b>_o_INTEGRATE LOG FILE</b>	<b>INTEGRATE MIRROR LOG FILE</b>	v16	Deprecated
コンパイラーテーマ:			
<b>_o_ARRAY STRING</b>	<b>ARRAY TEXT</b>	v12	廃止予定
<b>_o_C_GRAPH</b>	( <b>GRAPH</b> コマンドとSVGを併用して下さい)	v12	廃止予定
<b>_o_C_INTEGER</b>	<b>C_LONGINT</b>	v12	廃止予定
<b>_o_C_STRING</b>	<b>C_TEXT</b> (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:			
<b>_o_ADD SUBRECORD</b>	N->1 リレーションのテーブルにおいて <b>ADD RECORD</b>	v12	廃止予定
<b>Modified</b>	<b>Form event</b> と <u>On data change</u>	v12	廃止予定
<b>_o_MODIFY SUBRECORD</b>	N->1 リレーションのテーブルにおいて <b>MODIFY RECORD</b>	v12	廃止予定
フォームイベントテーマ:			
<b>_o_During</b>	<b>Form event</b> 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:			
<b>GRAPH</b> (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
<b>_o_GRAPH TABLE</b>	データを配列にビルドし、SVGピクチャー内で <b>GRAPH</b> を使用	v13	4D v14では無効化
階層リストテーマ:			
<b>_o_REDRAW LIST</b>	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:			
<b>_o_DISABLE BUTTON/_o_ENABLE</b>	<b>OBJECT SET ENABLED</b>	v12	廃止予定



## BUTTON

### ピクチャーテーマ:

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

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

### SQL テーマ:

<b>_o_USE EXTERNAL DATABASE</b>	<b>SQL LOGIN</b>	v12	廃止予定
<b>_o_USE INTERNAL DATABASE</b>	<b>SQL LOGOUT</b>	v12	廃止予定

### 文字列テーマ:

<b>_o_Convert case</b>	<b>CONVERT FROM TEXT/Convert to text</b> を必要に応じて使用して下さい	v11	廃止予定
<b>_o_ISO to Mac</b>	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
<b>_o_Mac to ISO</b>	(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
<b>_o_Mac to Win</b>		v11	廃止予定
<b>_o_Win to Mac</b>		v11	廃止予定
<b>サブレコードテーマ: 全コマンド</b>	“nnn SUBRECORD” と “nnn SUBSELECTION” 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
<b>システムドキュメントテーマ:</b>			
<b>Document type</b>	-	v12	廃止予定
<b>システム環境テーマ:</b>			
<b>_o_Font name</b>	フォント識別子を使用	v14	廃止予定
<b>_o_Font number</b>	QuickDraw の廃止に伴い、 <b>_o_Font name</b> と <b>_o_Font number</b> コマンドも廃止になります	v14	廃止予定
	<b>OBJECT SET FONT</b> 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。		
<b>ユーザーインターフェーステーマ:</b>			
<b>_o_Get platform interface/_o_SET PLATFORM INTERFACE</b>	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
<b>Windows テーマ:</b>			
<b>_o_Open external window</b>	4D 64-bit版ではサポートされていません	v16	廃止予定
<b>XML テーマ:</b>			
<b>_o_XSLT APPLY TRANSFORMATION</b>	PHP <i>libxslt</i> モジュールまたは <b>MissingRef</b> コマンドを使用して下さい	v14R4	v14R4では廃止予定
<b>_o_XSLT GET ERROR</b>	PHP <i>libxslt</i> モジュールまたは <b>MissingRef</b> コマンドを使用して下さい	v14R4	v14R4では廃止予定
<b>_o_XSLT SET PARAMETER</b>	PHP <i>libxslt</i> モジュールまたは <b>MissingRef</b> 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:

<code>_o_AP ShellExecute</code>	<b>LAUNCH EXTERNAL PROCESS</b>	v11	削除済み
<code>_o_AP Save BMP 8 bits</code>	"ピクチャ" テーマの4Dコマンドを使用して下さい	v14 R5	削除済み
<code>_o_AP FCLOSE、 _o_AP fopen、_o_AP FPRINT、_o_AP fread</code>	-	v14 R5	削除済み
<code>_o_AP Get file MD5 digest</code>	<b>Generate digest</b>	v14 R5	削除済み
<code>_o_AP BLOB to print settings</code>	<b>BLOB to print settings</b>	v16	廃止予定
<code>_o_AP Print settings to BLOB</code>	<b>Print settings to BLOB</b>	v16	廃止予定
<code>_o_AP Is picture deprecated</code>	<b>GET PICTURE FORMATS</b>	v16	廃止予定
<code>_o_AP NORMAL SCREEN、_o_AP FULL SCREEN</code>	-	v16	廃止予定
<code>_o_AP Get field infos、 _o_AP Get table infos</code>	-	v16	廃止予定
<code>_o_AP Get tips state、 _o_AP SET TIPS STATE</code>	-	v16	廃止予定

## 廃止予定コマンドの改名と非表示

4Dランゲージの分かりやすさのため、4D v15以降、廃止予定のコマンドには全て"`_o_`"が接頭辞として付けられる事になりました(まだ付けられていないものや4Dリスト(コードエディター、タイプアヘッド機能、他)に含まれていない場合)。

これらのコマンドは既存のコードからは削除されず、サポートされる限りは通常通り動作します。廃止予定のコマンドであっても、その名前に"`_o_`"を頭に付けることで廃止伊予定のコマンドをメソッドに追加することは可能です(ただし推奨されません)。このときこれらは正常に解釈されます。

## \_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。

# 🔧 GET PICTURE FORMATS

GET PICTURE FORMATS ( picture ; codecIDs )

引数	型		説明
picture	ピクチャー	➡	解析するピクチャーフィールドあるいは変数
codecIDs	テキスト配列	←	ピクチャーのコーデックID

## 説明

**GET PICTURE FORMATS** コマンドは、引数として渡された *picture* 引数内に含まれている全てのコーデックIDの配列を返します。4D ピクチャー (フィールドまたは変数) は、PNG、BMP、GIF など、複数の異なるフォーマットでエンコードされた同一の画像を格納することができます。

*picture* 引数には、含まれるフォーマットを *codecIDs* 配列内に取得したいピクチャーフィールドあるいは変数を渡します。

返されるコーデックIDは、**PICTURE CODEC LIST** コマンドと同様に4Dによって確立されます。これらは以下の形式で受け取ることが可能です:

- 拡張子(例: ".gif")
- Mimeタイプ(例: "image/jpeg")
- 4文字のQuickTimeコード

## 注:

- 4Dによって内部的に管理される以下のコーデックについては、必ず拡張子形式で返されます: JPEG、PNG、TIFF、GIF、BMP、SVG、PDF、EMF
- 4文字のQuickTimeコードは、QuickTime support 互換性オプションが (**SET DATABASE PARAMETER** コマンドを使用して) 設定されているデータベースにおいて返すことが可能です。しかしながら、QuickTimeは4Dではサポートされておらず、QuickTimeコーデックの使用は推奨されません。

ピクチャーコーデックIDについてのより詳細な情報については、**ピクチャ**の章を参照して下さい。

## 例題

カレントレコードのフィールド内に保存されているピクチャーフォーマットを知りたいという場合を考えます:

```
ARRAY TEXT($aTPictureFormats;0)
//保存されている全てのフォーマットを取得
GET PICTURE FORMATS([Employees]Photo;$aTPictureFormats)
```



## 🔧 \_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* 引数を使用した場合を除く、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)
```

# Maintenance & Security Center

-  概要
-  情報ページ
-  アクティビティ解析ページ
-  検証ページ
-  バックアップページ
-  圧縮ページ
-  ロールバックページ
-  復元ページ
-  修復ページ

## ⚙️ \_o\_Font number

\_o\_Font number ( fontName ) -> 戻り値

引数	型		説明
fontName	文字	→	フォント番号を返すフォント名
戻り値	倍長整数	↩	フォント番号

### 説明

---

このコマンドは廃止予定であり、4D v14以降使用されるべきではありません。このコマンドは互換性のために残されていますが今後のバージョンでサポートされなくなります。

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

\_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。

# 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	4D 64-bit版 での 状況
Mac OS QuickDraw フォント	フォント名	廃止 予定	削除 済み
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	削除 済み
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)	<b>WEB GET VARIABLES</b> コマンド(変数を取 得)、 <b>WEB GET BODY PART/WEB Get body part count</b> コマンド(ポストしたファ イルを取得)	廃止 予定	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	削除 済み
QuickTime サポート(互換性オプション は、 <b>SET DATABASE PARAMETER(QuickTimeSupport;1)</b> 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	廃止 予定	削除 済み
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	削除 済み
変換されたサブテーブル	N->1 テーブルの使用	廃止 予定	廃止 予定
XSLT	PHP <i>libxslt</i> モジュールまたは <b>PROCESS 4D TAGS</b> コマンドを使用して下さい	廃止 予定	削除 済み
4D Pack	統合された様々な4Dコマンド、あるいはその 他のテクノロジー	廃止 予定	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	OS	OS (cicn)

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

アイコン:  
削除済み)

PICT

新しいフォーマットを使用して下さい。詳細については[GET PICTURE FORMATS](#)を参照のこと

OS

削除済み



## 過去のドキュメント

---


このドキュメントは、**4D v16 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 v15](#) - (Rev. June 2015)
- [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)

## 廃止予定の機能と削除された機能 - 新着

16.0

15

 v16プロダクトレンジで廃止予定または削除された機能 Updated 16.0