

4D Compiler[®] 2.5

追加 / 修正情報

Windows[®] and Mac[™]OS Version

注意

1. このソフトウェアの仕様、およびマニュアルに記載されている事項を、将来予告なしに変更することがあります。
2. このソフトウェアおよびマニュアルのいかなる変更に関してもACI SAおよびACI USは、一切の責任を負いかねます。
3. このソフトウェアおよびマニュアルは、本製品のLicense Agreement（使用許諾契約書）のもとでのみ使用することができます。
4. このソフトウェアおよびマニュアルの一部または全部を、ACI SA/ACI USの明示的な記述による許可なしに、いかなる目的であれ、電子的、機械的、またどのような形であっても、無断で複製、配布することはできません。

©1985-1997 ACI SA/ACI US. All rights reserved.

4th Dimension、4D、4Dのロゴ、ACI、ACIのロゴは、ACI SAの登録商標または商標です。
その他、記載されている会社名、製品名は、各社の登録商標または商標です。

この文章は、下記の2つの項目から構成されています。

- ・ “ 詳細 ” には「4D Compilerリファレンスマニュアル」の内容を補足する情報があります。
- ・ “ 正誤表 ” には「4D Compilerリファレンスマニュアル」に関する訂正があります。

尚、このドキュメントには、Windows版の4D Compiler 2.5に関する情報も記述されていませんのでご参考にしてください。

詳細

4D Compiler2.5は、バージョン3.5の4th Dimensionおよびバージョン1.5の4D Serverと互換性があります。

システム必要条件

4D Compiler2.5を動作させるには、漢字Talk7以上が必要です。MacOS 7.5以前のシステムを使用している場合、Apple Computer,Inc.から提供される「Thread Manger」が4D Compilerのインストール中に自動的にインストールされます。

プロセッサの種類

バージョン2.5の4D Compilerから、コンパイル対象となるマイクロプロセッサの種類を選択できるようになりました。これにより、データベースのコンパイルを行うマシンではなく、コンパイル後のデータベースを動かすMacintoshまたはPCのモデルを選択することができます。実際に使用するプロセッサを特定して、コンパイルオプションを適切に選択すれば、プロセッサの特性を活かすことができます。選択肢は次のようになっています。

Motorola 68xxx (Macintosh)
Motorola PowerPC (Power Macintosh)
386/486およびPentium (Windows)

注：4D Compiler製品の中には、コンパイルオプションすべてを使用できないものもあります。

Motorola 68xxx (Macintosh)

68000用にコンパイルされたデータベースは、すべてのMacintosh上で動作します。68020と68030マイクロプロセッサは、基本的に68000より高性能です。3番目のオプション(「68020/68030 + 68LC040」または「680xxx + 6888X / 68040」)を選択すると、これらの機能を活用することができます。

注：68000用にコンパイルしたデータベースは、68000を搭載したMacintoshと68020や68030、68040を搭載したMacintoshでも動作します。しかし、68020/30/40用にコンパイルしたデータベースは68000を搭載したMacintoshでは動作しません。この場合、コンパイル後のデータベースを起動するとエラーメッセージが表示されます。

68020または68030プロセッサを搭載したMacintoshがコンパイルの対象機種である場合、数値演算用プロセッサ(6888X)はあってもなくても構いません。68040はボード上に数値演算プロセッサを備えています。68LC040は持っていません。オプションの中から適切なものを選択してください。

対象機種

CPU	コプロセッサ ?	コンパイルオプション
68000	使用せず	68000
68020または68030	なし	68020/68030 + 68LC040
68020または68030	あり(6888x)	680xx + 6888x / 68040
68LC040	使用せず	68020/68030 + 68LC040
68040	あり(ボード上)	680xx + 6888x / 68040

コンパイル後のデータベースを、複数モデルのMacintosh上で動かす場合(シングルユーザ)や、クライアントワークステーションに複数モデルのMacintoshを使用する場合は、すべてのMacintosh上でデータベースが動作できるように、コンパイルオプションとしては最下位のものを選択する必要があります。たとえば、数値演算用のコプロセッサを持たない68020/030のMacintoshクライアントが1台でも存在していれば、コプロセッサ有りのオプションは指定できません。「68020/030 + 68LC040」オプションを使用します。

68000以外のオプションを選択した場合は、32K以上のプロセス変数と32K以上のインタープロセス変数を使用できます。ただし、プロセス変数の設定には注意が必要です。プロセス変数がどのプロシージャで使われているか、コンパイル時に4D Compilerからは判断できません。プロセス変数には、プロセスごとに異なる値を持たせることができます。そこで、プロセスが新しく開始されるごとに全プロセス変数の複製が作られます。

たとえば、プロセス変数が50Kあるとすると、プロセスごとに50Kの追加メモリが必要になります。

また、プロセス変数に使われるメモリスペースは、プロセスのスタックサイズにはまったく関連がないことを覚えておいてください。

Motorola PowerPC (Power Macintosh)

このバージョンで生成されたPPCコードは、Power Macintosh™コンピュータに使用されているPowerPCプロセッサ601、603、604に対応しています。

このオプションを選択してコンパイルすると、動作環境は完全にPowerPC対応に最適化されます。4D Compilerによって生成されたネイティブコードは、PPC対応 (PPCネイティブまたはFAT) の4th Dimension、4D Client、4D Engineで実行させることができます。4D環境全体が、Power Macintoshの処理速度で動作します。

技術上の理由から、Power Macintoshでプロシージャあるいはスクリプトから、**New Process**関数を呼ぶ場合、プロセスに割り当てるスタックサイズを指示する数値引数には偶数を使用する必要があります。

たとえば、次のように指定してください。

```
̄CustProc:=New Process("mDoCust"; 32000; "Customers")
```

次のような指定はできません。

```
̄CustProc:=New Process("mDoCust"; 32767; "Customers")
```

PC (Windows)

コンパイル後のデータベースを4D ServerとさまざまなPCクライアントとともに使用する場合は、最も高性能なコンパイルオプションを選択することができます。Pentiumと486のクライアントがある場合は、「Pentiumの最適化」を選択できます。

マルチプラットフォーム向けのコンパイル

コンパイルを行っているプラットフォーム以外のプラットフォームで動かすためのデータベースをコンパイルするには、対応する4D Compilerプラットフォームエクステンションを入手する必要があります。Windows上でコンパイルを行い、コンパイル後のデータベースをMacintoshやPower Macintoshで動作させるには、コンパイル前に、対応するMacintosh/Power Macintoshのエクステンションを「Mac4DX」ディレクトリまたは「Proc.ESR」ファイルに入れておく必要があります。同様に、Macintosh上でコンパイルし、コンパイル後のデータベースをWindows上で動作させる場合は、Windows用のエクステンションを「Win4DX」フォルダまたは、「Proc.Ext」ファイルに入れておく必要があります。

2種類の異なるプラットフォーム上で動作するように設計されたデータベースをコンパイルするためには、ストラクチャと同じ階層に「Mac4DX」と「Win4DX」フォルダ/ディレクトリを置く必要があります。

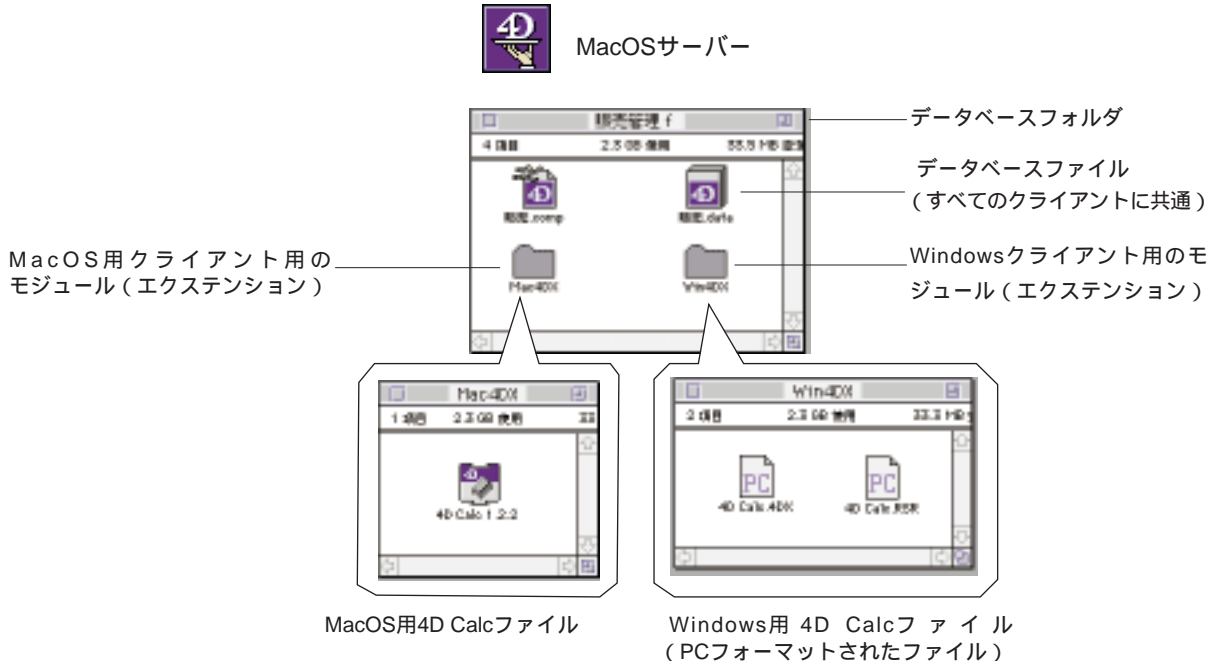
注：4D Compilerバージョン2.5は従来のエクスターナル方式にも対応しています。データベースストラクチャに納められたり、「Proc.Ext」ファイルに入っている外部ルーチンも正しく認識します。

MacOS上のマルチプラットフォーム向けのコンパイル

(4th Dimensionや4D Compilerを使って) Windowsプラットフォーム上で実行されている4Dエクステンションを使用するMacOS上のデータベースをコンパイルするには、次のように行います。

1. Windows版の4DエクステンションをPCマシン上にインストールする。
2. MacintoshまたはPower Macintoshの中にそのWindows版のエクステンションファイルをコピーする。
これらのWindows版のエクステンションは、“エクステンション.4DX”と“エクステンション名.RSR”ファイルを持っています。
3. MacOSのマシンに“Win4DX”というフォルダを新規に作成する。
4. 作成した「Win4DX」フォルダの中にWindows版のエクステンションファイルをコピーする。
5. 「Mac4DX」フォルダの次にコンパイルするデータベースのストラクチャファイルがある同じ階層にこのフォルダを配置する。

下図は、上記の手順を4D Calcを例にとって示したものです。



注：プラットフォームに依存しない4D Serverに関する詳細は、4D Serverマニュアルを参照してください。また、「Mac4DX」フォルダと「Win4DX」ディレクトリ (フォルダ) の中には、同一バージョンの4Dエクステンションを入れる必要があります。

Windows上のマルチプラットフォーム向けのコンパイル

外部ルーチンを使用しているデータベース

680xx用の外部ルーチンを使用しているデータベースをWindows上でコンパイルするには、4D Transporterを使ってProc.Extファイルをトランスポートする必要があります。トランスポートの結果は、“ Proc.ESR ” という名前のファイルになります。このファイルを4D Compilerで認識できるようにストラクチャと同じ階層に置く必要があります。

4Dエクステンションを使用するデータベース

MacOSプラットフォーム上で実行されている4Dエクステンションを使用するWindows上のデータベースをコンパイルするには、次のように行います。

1. MacintoshまたはPower Macintoshの4DエクステンションパッケージをMacOSのマシンにインストールする。

注：後にまたMacOS上でそのデータベースを実行する場合、MacOSの4Dエクステンション名は、3バイトの接尾語（拡張子）を持った8バイトの名前にしておく必要があります。もし、このルールに従わないと、適当な名前に変更されます。データベースを実行中、そのエクステンション名はコンパイル中の名前と同じでなければなりません。

2. 4D Transporterを使って、その4DエクステンションをWindows用にトランスポートする。4D Transporterに関する詳細は「ユーティリティガイド」オンラインドキュメントを参照してください。

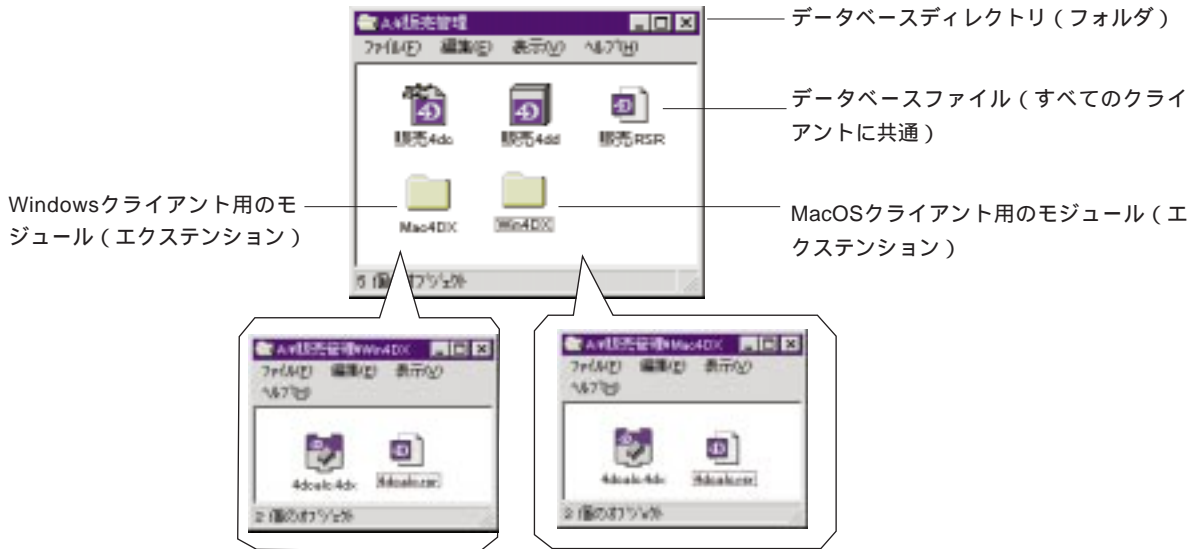
注：トランスポートされたエクステンションは、“ エクステンション名.4DX ” と “ エクステンション名.RSR ” ファイルを持っています。

3. “ Mac4DX ” というディレクトリ（フォルダ）を新規に作成する。
4. コンパイルするデータベースのストラクチャがある同じ階層にこのディレクトリを配置する。

下図は、前記の手順を4D Calcを例にとって示したものです。



Windowsサーバー



Windows用4D Calcファイル トランスポートされたMacOS用4D Calcファイル

外部ルーチンによって作られた変数

外部ルーチンで作成される変数をコンパイル中に4D Compilerから認識させるための唯一の方法は、データベースストラクチャ内のコーディングによって宣言しておくことです。

MacOS

こうした変数を4D Compilerに確実に認識させるには、データベースストラクチャのファイル内にResEdit™などのリソースエディタを使用してリソースを作成する必要があります。変数がプロセス変数の場合、“VAR#”リソースにします。変数がインタープロセス変数の場合、“VAR<>”リソースにします。これらリソース内に変数を宣言します。

Windows

“VAR#”と“VAR<>”リソースが存在すると、それらは認識され、Windowsプラットフォーム上でもMacOSプラットフォームの場合と同じように利用されます。データベースにこれらのリソースが必要な場合、MacOSプラットフォーム上でResEdit™などのリソースエディタを利用して作成することができます。

Windows上の「バージョン番号自動生成」オプション

Windows上でバージョン番号を調べるには、**SET ABOUT**コマンドを使用します。これにより、4th Dimension、4D Compiler、コンパイル後のデータベースのバージョン番号を表示させることができます。

4D Compiler Proによる実行形式のアプリケーションの作成

Windows

コンパイルの開始前に、以下のファイルをハードディスクにコピーし、同じディレクトリ内で使用できるようになっていることを確認してください。

4DENGINE.4DE	4DENGINE.RSR
ASIFONT.FON	ASIFONT.MAP
ASINTOPPC.DLL	ASIPORT.RSR
SERIAL16.DLL	

「アプリケーション作成（4D Engineを組み込む）」オプションをクリックすると、「ファイルオープン」ダイアログボックスが表示されます。

4D Engineを選択し、「開く」ボタンをクリックします。「メイン」ウインドウに戻ると、選択した4D Engineが表示され、実行形式を作成するという選択結果を示します。

コンパイル時には、4D Engineとコンパイル後のストラクチャから実行形式のアプリケーションが作成されます。Windowsでは、コンパイラから「データベース名.EXE」、「データベース名.4DC」、「データベース名.RSR」の3種類のファイルが出力されます。

「データベース名.EXE」は実行形式のアプリケーション、「データベース名.4DC」と「データベース名.RSR」はコンパイル後のアプリケーションです。

注：「アプリケーション作成（4D Engineを組み込む）」オプションを利用した場合、4th Dimension、4D Server、4D FIRSTで次のファイルをオープンできなくなります。

データベース名.EXE
データベース名.4DC
データベース名.RSR

実行形式のアプリケーションをインストールするには、以下の要素を同じディレクトリに置きます。

- ・データベースファイル
- ・Win4DXディレクトリ（エクステンションルーチン使用の場合）
- ・オプションで、CTL3D32.DLLとCTL3D.DLL

アプリケーションを配布するには、Stirling Technologies社のInstallShieldなどの、一般に販売されているインストール用プログラムを利用できます。この種のインストーラには次のような利点があります。

- ・データベースのユーザが、カスタムインストールか標準のインストールを選択できる。
- ・Microsoft社のWin32などのソフトウェアが必要かどうか決定し、自動的にインストールできる。
- ・アプリケーションアイコンをプログラムマネージャに登録できる。
- ・複数ディスクにわたるインストーラを作成できる。

4D Windows版での実行

コンパイルしたデータベースをWindows版の4th Dimensionまたは4D Engineで開くと、“データベース名.CMP”と命名されたファイルが、コンパイルしたストラクチャと同じ階層に作成されます。このファイルは、コンパイルされたデータベースを初めて開く際に作成されるもので、そのデータベースを起動するために必要なファイルです。

OLE外部プロシージャを使用したコンパイル

データベースでOLE外部プロシージャを使用している場合は、コンパイルを行う前に以下の手順をとる必要があります。

- ・ “Win4DX” という名前のディレクトリを新規に作成する。
- ・ OLETOOLS.DLLファイルを “OLETOOLS.4DX” という名前に変更する。
- ・ OLETOOLS.4DXとOLETOOLS.RSRの2ファイルをWin4DXディレクトリに配置する。

これで、データベースをコンパイルできるようになります。

コンパイル後のデータベースの起動

1. OLETOOLS.DLLとOLETOOLS.RSRファイルの入ったWin4DXディレクトリをそのままにしておくか、または
2. OLETOOLS.DLLとOLETOOLS.RSRを4D.EXEファイルと同じディレクトリに配置して使用することもできます。

実行形式のアプリケーションの起動

Win4DXディレクトリは、OLETOOLS.4DXとOLETOOLS.RSRとともに保持しておく必要があります。

正誤表

エラーメッセージ

- ・メッセージ：
“ ローカル変数の合計サイズが32KBを超えました。サイズ：xxバイト ”
これは、Macintosh、Windowsいずれのプラットフォームにも適用されます。
- ・メッセージ：
“ プロシージャのサイズが32KBを超えました。”
これは、Macintoshのみに適用されます。「付録A-11ページ」

その他

- ・下記のボタンラベルが変更されました。
「メイン」ウインドウ内の「続く…」ボタン 「次…」ボタン 「2-8、2-14ページ」
「4D Compiler」ウインドウ内の「中断」ボタン 「停止」ボタン 「2-20ページ」
「4D Compiler」ウインドウ内の「終了」ボタン 「中止」ボタン 「2-20ページ」
- ・「ファイル」メニュー内の下記のメニューアイテムタイトルが変更されました。
「名前を変えて保存」 「新規保存」 「2-6ページ」
「元に戻す」 「復帰」 「2-6ページ」
- ・マニュアルの中で例題として使用されているデータベースの名前が「シミュレーション」から「デモ」に変更されました。尚、内容はまったく同じです。
「1-9、1-11」ページ

