

OCI Mapper 2004-3

By Josh Fletcher, Technical Support Engineer, 4D, Inc.

TN 06-06

Abstract

このテクニカルノートには、過去のテクニカルノート 03-44 “Migrating from 4D for Oracle to 4D for OCI using OCI Mapper”で紹介された OCI Mapper コンポーネントのアップデート版が付属しています。本文中、同コンポーネントに加えられた修正点、4D for OCI プログラミングに関する考察、および 4D for Oracle との相違点などの論題が扱われています。OCI Mapper コンポーネントのインストール手順についても説明されています。

このテクニカルノートは、前後編から成る OCI Mapper アップデートノートの前編です。後編では、OCI Mapper のデバッグ版が提供されます。デバッグ版では、4D for OCI コールや OCI Mapper メソッドのログをテキストファイルに記録する機能が加えられています。これは 4D for OCI プログラミングの解析に役立つはずですが、デバッグコードはメソッドが煩雑になるため、分けて公開することにしました。

Introduction

このテクニカルノート（および OCI Mapper コンポーネント）の目的は次のとおりです：

- 4D for Oracle 使用データベースの場合、OCI Mapper を使用することによってソースコードを変更することなく、スムーズに 4D for OCI への移行を果たすことができます。
- 4D for OCI（あるいは OCI 全般）プログラミングに初めて取り組むデベロッパの場合、OCI Mapper によって 4D for OCI には存在しないハイレベルコマンドによるフレームワークが提供されます。

4D for Oracle デベロッパにとって、このテクニカルノートは、4D for Oracle と 4D for OCI コマンドの良い比較材料であり、OCI Mapper コンポーネントを経て 4D for OCI プログラミングへ移行するための導入となるはずです。

4D for OCI デベロッパにとって、OCI Mapper はハイレベルで API 的なフレームワークを OCI 開発環境に加える例として参考になるはずです。

OCI Mapper のアップデートに際しては、バグの修正と共に 4D for Oracle との違いを際立たせることに重点が置かれました。

OCI Mapper はオープンソースで提供されており、デベロッパは自由にコードを変更すること

ができます。OCI Mapper のコードに加えられた修正の大部分は、コードの中でドキュメント化されており、コメントアウトされたオリジナルコードと共に新しいコードと変更理由が記述されています。あるものは単純なバグ修正ですが、多くのものは典型的な落とし穴を際立たせるため、つまり 4D for OCI プログラミングではいけない例を例示するためのものです。

内容の重複を避けるため、4D for Oracle と 4D for OCI のパフォーマンス比較はこのテクニカルノートでは扱っていません。プラグインのベンチマーク比較については、テクニカルノート 03-44 “Migrating from 4D for Oracle to 4D for OCI using OCI Mapper”を参照してください。

Overview of 4D for Oracle and 4D for OCI

03-44 “Migrating from 4D for Oracle to 4D for OCI using OCI Mapper”からの再録

4D for Oracle と 4D for OCI は、どちらもコネクティビティプラグインで、4th Dimension が Oracle データベースと通信するための外部ルーチン群を提供しています。Oracle データベースに保存されたデータを表示、操作、更新することが主な機能で、4th Dimension の快適なユーザインタフェース開発環境と合わせれば、Oracle データベースのフロントエンドアプリケーションを作成する上で強力なツールとなるという点が共通しています。

しかし、両プラグインは 4D および Oracle のサポートレベルに違いがあります：

サポート対象	4D for Oracle	4D for OCI
4D 2004	×	○
ハイレベルコマンド	○	×
ローレベルコマンド	一部	○
OCI ドライバ/Oracle Client	バージョン 7.x/8	バージョン 8.1.6 以降

4D 2004 では、4D for OCI だけがサポートされる点に注意してください。これは、4D for Oracle のハイレベルコマンドがもはや使用できないことも意味します。

Advantage of using 4D for OCI over 4D for Oracle

4D for Oracle をやめて 4D OCI を使用する理由は？

もっとも直接的な理由は、4D for Oracle がバージョン 2004 では存在しないため、というもののですが、その他にも充分な理由があります。

4D for Oracle では、特定のタスクを意図した多数のハイレベルおよびローレベルコマンドが提供されていた反面、コマンドの用法が固定的で、デザインの自由という面では限界がありました。

これに対し、4D for OCI では極めてローレベルなコマンドが提供されており、カスタマイズされたソリューションを作成できる可能性を秘めています。4D for OCI は柔軟性があり、ローレベルであるため、デベロッパは Oracle プログラミングを完全にコントロールすることができます。

テクニカルノート 03-44 で証明されているように、4D for OCI は 4D for Oracle よりずっと高速に動作します。

Where does the OCI Mapper fit in?

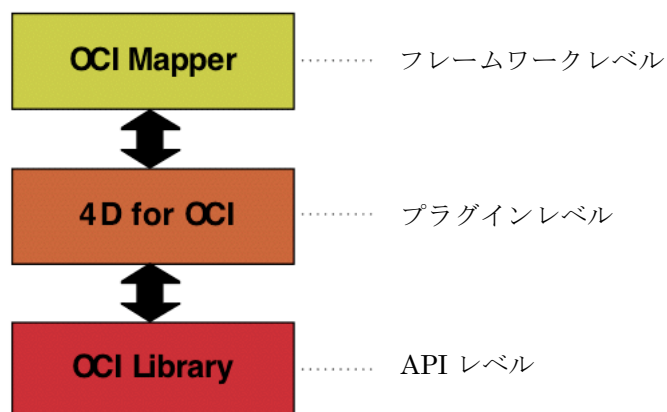
4D for OCI の利点は、同時に難点でもあるかもしれません。つまり、すべてローレベルコマンドであるため、コーディングに大変な労力を要します。理想は、4D for OCI コマンドをラップして、目的のデザインに合った API 的フレームワークを設けることでしょう。OCI Mapper はまさにそのようなフレームワークを提供します。

Introduction to the OCI Mapper

OCI Mapper は、4D for Oracle のハイレベルコマンドをエミュレートして動作するフレームワークです。コードは 4D for OCI コマンドとネイティブの 4D ランゲージで作成されており、4D コンポーネント（ソースコード付き）の形式で提供されています。

Architecture

Oracle 開発環境における OCI Mapper の位置づけを図に表わしてみました：



この構図では、フレームはデベロッパ、プラグインは 4D 社、API は Oracle 社が提供します。

OCI Mapper 2004-3 in-depth

OCI Mapper 2004-3 では、オリジナルのデザインの多数の修正が加えられました。ここでは、今回のアップデートの主要な内容、4D for Oracle および 4D for OCI デベロッパ向けの重要な情報などがまとめられています。

4D for Oracle Developers

4D for Oracle デベロッパが OCI Mapper を使用する最大の目的は 4D 2003 から 4D 2004 に移行することであることに配慮し、4D for Oracle から 4D for OCI に移行する場合はデータベース内のコードを修正する必要がないようになっています。OCI Mapper は 4D for Oracle と同じコマンド名を使用し、同じ動作をエミュレートしています。この結果、最少限の労力で以前の 4D for Oracle データベースをアップデートし、運用することができます。

OCI Mapper には 4D for OCI と 4D for Oracle プログラミングの相違を際立たせるという目的もあり、研究材料的な側面もあるため、完全なソリューションというよりは、参考資料程度にみなされるべきです。

4D for OCI Developers

4D for OCI デベロッパの場合、OCI Mapper は 4D for OCI コマンドをラッピングして、有用なハイレベルコマンドからなるフレームワークを作成する実例となっています。

この場合も、OCI Mapper はソリューションではなく、例題とみなされるべきです。

Important Notes For OCI Mapper 2004-3

ここでは OCI Mapper に関する重要な論題が取り上げられています。

Concurrent Cursors

OCI Mapper では、同時複数のカーソル管理（配列でカーソルの状態を管理）はサポートしていますが、複数の出力パラメータはサポートしていません。

例えば、**cursor1** と **cursor2** がある場合：

- **cursor1** の出力パラメータを定義
- **cursor2** の出力パラメータを定義
- **cursor1** を実行
- 結果をフェッチ

上記のような操作をすると、結果は **cursor1** ではなく、**cursor2** の出力パラメータにフェッチされます。したがって、次のように処理する必要があります：

- **cursor1** の出力パラメータを定義
- **cursor1** を実行
- 結果をフェッチ
- **cursor1** を閉じる
- **cursor2** について同じことを繰り返す

あるいは、複数の出力パラメータをサポートするように OCI Mapper のコードを変更しても構いません。

Error Handling

OCI Mapper は、4D for OCI コールをした後のエラー処理が完全ではありません。OCI Mapper コンポーネント自体のデバッグが目的なのであれば、そのためのエラーハンドリングを追加する必要があります。

デバッグが目的なのであれば、OD ON ERR CALL コマンドによるエラーハンドリング (CheckError メソッドがそのまま利用できます) をすると良いでしょう。このコマンドにより、4D for OCI エラーはすべてトラップすることができます。あるいは、導入部分で触れたテクニカルノート後編のデバッグ版 OCI Mapper を利用するという方法もあります。

もちろん、OCI Mapper のコードを書き換えて、本格的なエラーハンドリングを組み込んでも構いません。

LOBs

OCI Mapper は、現時点で Large Object (LOB) タイプをサポートしていません。LOB は、4D の BLOB とピクチャ、Oracle では RAW、LONG RAW、BLOB、CLOB、NCLOB に相当します。

OCI プログラミングでは、LOB データを単純にステートメント実行の中でバインドやフェッチすることはできません。ステートメント実行後、OCI コードを続けて LOB 操作をする必要があります。(基本的に単純作業で LOB データをコピーすることを意味します。)

OCI Mapper に LOB サポートを付加するのであれば、OD BIND TOWARDS 4D (出力パラメータの定義)、OD BIND TOWARDS SQL (入力パラメータの定義)、OD EXECUTE CURSOR (ステートメントの実行)、OD Load rows cursor (データのフェッチ) メソッドの内容を把握しておくとい良いでしょう。

Important Method Notes for OCI Mapper 2004-3

ここには、OCI Mapper 2004-3 の特定のメソッドに関する重要な情報がまとめられています：

Method: **CheckError**

OCI Mapper が 4D for OCI エラーの処理をするエラーハンドリングメソッドです。OD ON ERROR CALL コマンドでカスタムエラーハンドリングメソッドにすることもできます。今回、OCI Mapper 2004-3 用に完全に書き直されました。主要な変更点は次のとおりです：

- すべてのエラーについてメッセージが用意されました。
- OCI エラーハンドリングは複数のエラーレコードを含むことができるので、新しい CheckError は詳細を扱えるようになりました。
- OD Last Error を活用するコードが追加されました。
- 4D for OCI を直接コールすることはなくなりました。

(汎用的な `oci_tool_GetErrorInfo` を利用しています。)

OCI エラーハンドリングに慣れていないのであれば、CheckError メソッドの内容が参考になると思います。

Method: **OD BIND TOWARDS 4D**

4D for Oracle デベロッパに対する注記：ここで **BIND** という語の用法は不正確です。OCI プログラミングでは、出力パラメータを定義するのであり、バインドするものではありません。厳密に言えば、このメソッドは **OD DEFINE OUTPUT PARAMETERS** ということになります。

Method: **OD Clone 4D Table**

4D テーブルのコピーを Oracle に作成するコマンドですが、幾つか考慮すべて重要な点があります：

- サブテーブルはサポートされません。
- BLOB やピクチャタイプのオプションは存在しますが、現時点では OCI Mapper でサポートされていません。(前述の LOB に関する注記を参照してください。)

Method: **OD Cursor state**

以前の OCI Mapper では、カーソルの処理に問題があったため、動作が不安でしたが、その問題は修正されました。

Method: **OD Execute object**

このメソッドは、Oracle パッケージに配置されたオブジェクトはサポートしていません。
パッケージの中にアクセスするためには、追加の OCI コーディングが必要です。

Method: **OD GET SERVER LIST**

このメソッドは、"tnsnames.ora"ファイルからエントリーを取得するために使用されます。Oracle Home ディレクトリの場所を知るために 4D for OCI のメソッドである OCIGetTnsnamesPath を使用していますが、Oracle 10g では ORACLE_HOME レジストリキーの場所が変更されたため、メソッドが正しい値を返さないという問題があります。

この問題を回避するためには、別の方法で Oracle Home を取得する、あるいは正しい値でレジストリキーを更新しなくてはなりません。Oracle 10g におけるレジストリキーは下記の場所にあります：

HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\ORACLE_HOME

Method: **OD Last Error**

このメソッドは、以前の OCI Mapper ではまったく動作していませんでしたが、今回、修正されました。詳細については CheckError のヘッダコメントを参照してください。

注記：このメソッドにおける"error"とは 4D for OCI コールの返り値が OCI_ERROR であることを意味します。OCI_NO_DATA や OCI_SUCCESS_WITH_INFO この条件を満たさないため、OD Last Error ではエラーとして扱われません。

4D for Oracle デベロッパに対する注記：4D for Oracle と互換性があるパラメータは "message"だけです。4D for Oracle に存在した他のパラメータは 4D for OCI にはありません。エラーコードも 4D for Oracle とは異なります。エラーコードについては Oracle のドキュメントを参照してください。例：ORA-00936 はエラーコード 936 です。

Method: **OD Login**

4D for Oracle デベロッパに対する注記："mode"パラメータは存在しません。

Method: **OD ON ERROR CALL**

4D for Oracle デベロッパに対する注記：カスタムエラーハンドルに返されるパラメータは大きく異なります。4D for Oracle のパラメータの中には、4D for OCI に存在しないものもあります。4D for OCI では存在しないパラメータは次のとおりです：

\$1	倍長整数	エラーの誘因となったオブジェクトの ID
\$2	倍長整数	エラーコード
\$3	テキスト	エラーの説明
\$4	倍長整数	エラーの発生元 (100;200;300)
\$5	倍長整数	コマンドテキストの先頭から数えたエラーの位置、
\$6	テキスト	エラーの誘因となったコマンド名

上記中、\$1、\$5、\$6 は 4D for OCI に存在しません。

エラーハンドリングメソッドは、次のようにふたつパラメータを受け取るように作成する必要があります：

- 、
- C_LONGINT(\$1)** エラーを報告した関数 (OCI_ERROR、OCI_INVALID_HANDLE) などからの返り値
- C_LONGINT(\$2)** OCI エラーハンドル (エラーハンドルには、実際のエラーデータを持っているエラーの記録が納められています。)

パラメータの詳細な使用法については、**CheckError** のソースコードを参照してください。

Method: **OD SET OPTIONS**

4D for Oracle デベロッパに対する注記：4D for Oracle のオプションの中には、OCI Mapper で利用できないものもあります：

オプション	値	説明
kNoWait	4	コンテキスト用。kWithLock オプションが選択されたコンテキストを開こうとした際ロックに遭遇した場合、エラーを返すことを指定する。その場合、クエリは SELECT...FOR、UPDATE OF...NOWAIT タイプ。指定しない場合、クエリはロックが解除されるまで待機する。
kWithLock	8	コンテキスト用。指定した場合、コンテキストは結果行をロックする。少なくともひとつのバインドが kForUpdate オプションを指定されている場合、クエリは SELECT...FOR、UPDATE OF... タイプ。
kDistinctLines	16	コンテキスト用。 SELECT 句に DISTINCT キーワードを追記し、コンテキストの結果行から重複を除くことを指定する。コンテキストは自動的に読み取り専用になる。
kDeferred	64	Oracle7 サーバにリクエストを送信する際、OCI の遅延モード使用を許可する。

OD SET OPTIONS で設定されるオプションは、Mapper ではグローバルに適用され、4D for Oracle のようにオブジェクト毎に設定することはできません。

OCI Mapper 2004-3 Command Reference

ここに列挙されているのは、OCI Mapper のハイレベルコマンドだけです。ローレベルコマンドについては、OCI Mapper コンポーネントのソースコードに含まれているドキュメントを参照してください。

コンポーネントのコマンドには、すべてヘッダコメントが記述されており、コマンドの目的、入出力パラメータが必要に応じて説明されています。

OD BIND TOWARDS 4D

出力パラメータを定義するために使用します。

OD BIND TOWARDS SQL

入力パラメータをバインドするために使用します。

OD Clone 4D Table

4th Dimension のテーブルと同じ構造の Oracle テーブルを作成するために使用します。

OD COMMIT

任意の接続に対して OCITransCommit をコールするために使用します。

OD Create cursor

特定の接続に対してカーソルを作成するために使用します。

OD Cursor state

特定のカーソルの状態を知るために使用します。

OD DROP CURSOR

OD Create cursor で作成したカーソルが使用していたメモリを解放するために使用します。

OD EXECUTE CURSOR

カーソルに関連付けられた SQL ステートメントを Oracle サーバで実行するために使用します。

OD Execute object

Oracle パッケージ以外の場所にあるストアードプロシージャまたはストアード関数を実行するために使用します。

OD Execute SQL

ログイン ID を指定し、SQL クエリを送信して結果を 4th Dimension のフィールド、変数、配列に保存するするために使用します。

OD GET COLUMN ATTRIBUTES

カーソルに関連付けられた SQL クエリの結果列のタイプとサイズを取得するために使用します。

OD Set SQL in cursor でカーソルに SQL ステートメントをセットした後に実行できます。

OD Get column title

OD Set SQL in cursor の後に実行し、結果列のタイトルを取得するために使用します。

OD Get NB Mode

非ブロックモードのカレントの状態を知るために使用します。

OD Get options

OCI Mapper 用のグローバルオプションに表わす倍長整数を取得するために使用します。

OD GET SERVER LIST

tnsnames.ora ファイルからサーバ定義のリストを配列に取得するために使用します。

OD Last Error

最後の 4D for OCI エラーの値を取得するために使用します。エラー番号についてはターゲットサーバの Oracle のドキュメントを参照します。

OD Load rows cursor

OD EXECUTE CURSOR で実行した SQL の結果行を配列で処理するために使用します。

OD Login

ログインパラメータを渡して Oracle サーバに接続し、接続 ID を取得するために使用します。

OD Login dialog

接続ダイアログを表示し、ユーザに Oracle サーバを特定させるために使用します。

OD Login state

特定の接続 ID が開かれた接続であるかを調べるために使用します。

OD LOGOUT

特定の接続を終了するために使用します。

OD Number of columns

クエリの結果列の数を取得するために使用します。OD Set SQL in cursor の後に実行します。

OD Number rows processed

OD Load rows cursor の実行後、SELECT 系のクエリでロードされた行の数、または INSERT、UPDATE、DELETE 系のクエリで挿入、更新、削除された行の数を取得するために使用します。

OD ON ERROR CALL

エラーハンドリングメソッドをインストールするために使用します。

OD ROLLBACK

Oracle トランザクションをキャンセルするために使用します。

OD SET NB MODE

非ブロックモードを設定するために使用します。

OD SET OPTIONS

OCI Mapper の動作についてグローバルオプション設定をするために使用します。

OD Set SQL in cursor

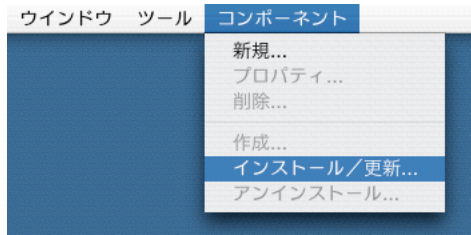
SQL ステートメントをカーソルに関連付けるために使用します。ステートメントは Oracle サーバに送信され、OD EXECUTE CURSOR をコールすることによって実行されます。

Installation Procedures for OCI Mapper 2004-3

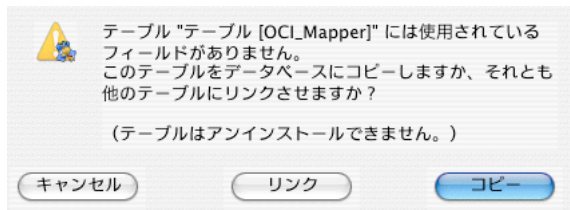
OCI Mapper 2004-3 を使用するためには 4D 2004 および 4D for OCI 2004 が必要です。

データベースに新しく OCI Mapper をインストールする

1. データベースを 4D Insider で開きます。
2. コンポーネントメニューからインストール／更新を選択します。



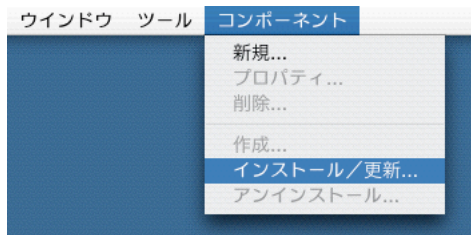
3. "OCI Mapper 2004-3.4CP"ファイルの場所をブラウズし、開くボタンをクリックします。
4. 4D Insider が OCI Mapper コンポーネントをインストールします。このとき、"OCI_Mapper"テーブルもコピーするようにしてください。



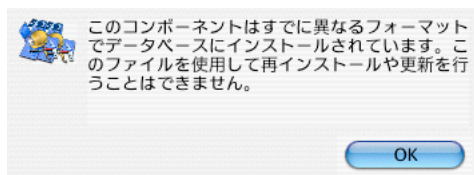
5. 4D Insider を終了します。
6. データベースを 4D で開きます。
7. On Startup データベースメソッドに OCI_TOOL_INITVAROCI を挿入します。
8. 4D for OCI プラグインが正しくインストールされていることを確認します。

以前のバージョンから OCI Mapper をアップグレードする

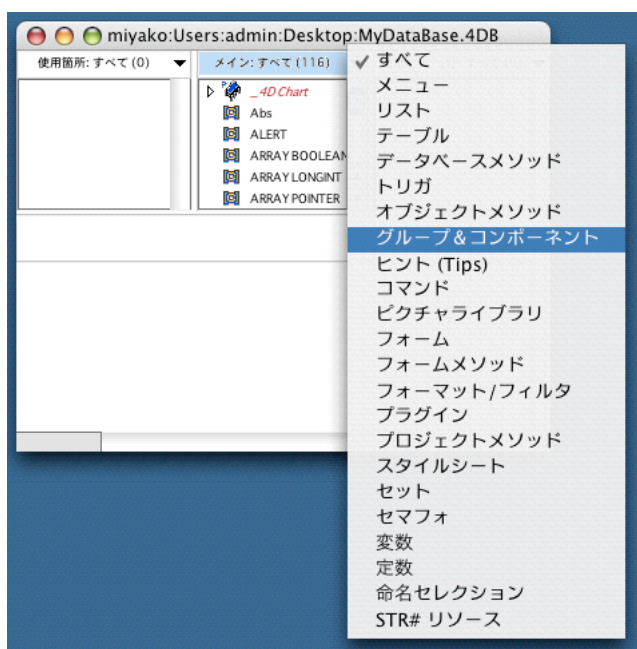
1. データベースを 4D Insider で開きます。
2. コンポーネントメニューからインストール/更新を選択します。



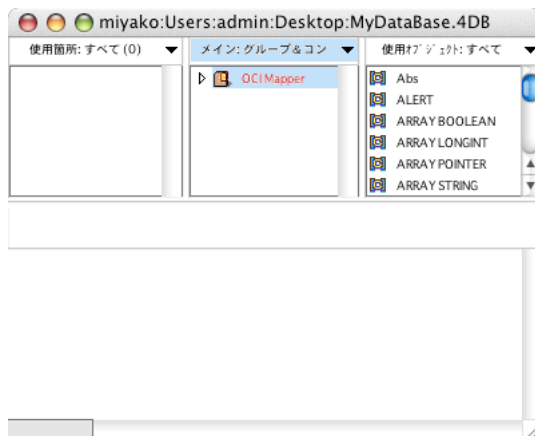
3. "OCI Mapper 2004-3.4CP"ファイルの場所をブラウズし、開くボタンをクリックします。
4. 4D Insider が OCI Mapper コンポーネントをインストールします。このとき、次のエラーが表示されたら次のステップ 5 に進みます：



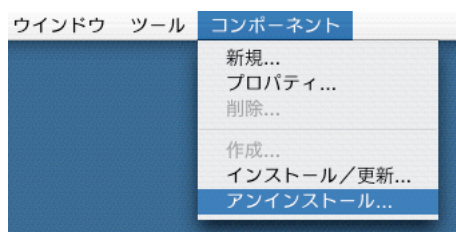
5. 先に古いバージョンの OCI Mapper をアンインストールする必要があるので、4D Insider のメインポップアップメニューをクリックし、グループ&コンポーネントを選択します：



6. OCI Mapper コンポーネントを選択します。



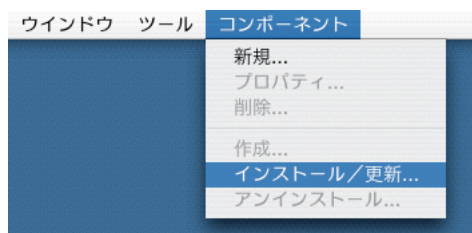
7. コンポーネントメニューからアンインストールを選択します：



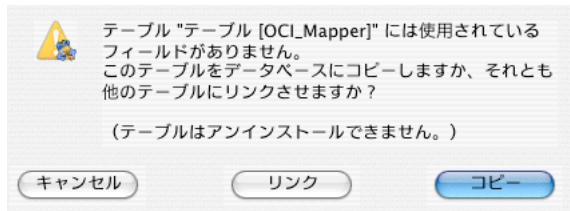
8. OK ボタンをクリックして OCI Mapper をアンインストールします。
9. ステップ 1 から 4 を再実行して、新しいバージョンの OCI Mapper をインストールします。

4D for Oracle データベースを OCI Mapper データベースに変換する

1. データベースを 4D Insider で開きます。
2. コンポーネントメニューからインストール/更新を選択します。



3. "OCI Mapper 2004-3.4CP"ファイルの場所をブラウズし、開くボタンをクリックします。
4. 4D Insider が OCI Mapper コンポーネントをインストールします。このとき、"OCI_Mapper"テーブルもコピーするようにしてください：



5. 4D Insider を終了します。
6. データベースを 4D で開きます。
7. 4D for Oracle コマンドを使用するすべてのメソッドを再トークナイズします。
8. 4D を終了します。
9. Win4DX/Mac4DX フォルダにインストールされた 4D for Oracle プラグインを取り除きます。
10. 4D for OCI プラグインをインストールします。
11. On Startup で OCI_TOOL_INITVAROCI メソッドをコールし、OCI Mapper を使用するプロセスの最初で oci_tool_Initprocvar コールするようにします。

Useful Resources

役に立つ 4D 情報

4D for Oracle ドキュメント :

ftp://ftp.4d-japan.com/ACI_PRODUCT_REFERENCE_LIBRARY/68/ORACLE67.pdf

4D for OCI ドキュメント :

ftp://ftp.4d-japan.com/ACI_PRODUCT_REFERENCE_LIBRARY/68/4DOCI8.pdf

役に立つ Oracle 情報

Oracle Call Interface プログラマーズガイド、10g リリース (10.2) :

http://downloadwest.oracle.com/docs/cd/B19306_01/appdev.102/b14250/toc.htm

Oracle Call Interface プログラマーズガイド、リリース 8.1.6 :

http://download-west.oracle.com/docs/cd/A87862_01/NT817CLI/index.htm

Conclusion

OCI Mapper は、4D for Oracle から 4D for OCI への移行を容易にするもので、4D for OCI による開発の時間と費用を大幅に削減することができます。OCI Mapper は生産性と効率を上げるために 4D for OCI 用のフレームワークを作成するという手法の良い実例です。