

4D v11 SQLでデータベースを複製する方法

By Silvio Belini, Technical Services Team Member, 4D Inc.

Technical Note 08-34

Abstract

このテクニカルノートでは、4D v11 SQLでうまくデータベースを複製するために正しい手順を説明します。ここでは、4D v11 SQL の2つのインスタンス間で、新しいXMLのストラクチャの読み込み/書き出し機能とドラッグ&ドロップのメカニズムを使用します。

Introduction

4D v11 SQLには、新しいドラッグ&ドロップのメカニズムが構築されています。そのため4D v11 SQL の2番目のインスタンスで起動中のデータベースを部分的にドラッグ&ドロップすることにより、デベロッパはデータベースを部分的に再度使用することができ、開発プロセスでの仕事量を低減することが可能です。

しかし、新たに作成されたデータベースで十分にその機能を得るには、いくつかのステップを踏まなければなりません。ここでは、その手順を説明します。

4D v11 SQLインスタンス間でのオブジェクト移動の手順

新たに作成されたデータベースがオリジナルデータとして機能するためには、オブジェクト間での参照は完全な状態を保っていなければなりませんし、テーブルとフィールドの内部的な採番も同様にそのままの状態でなければなりません。以下のステップを順番通りに実行することにより、上記で説明した特性を保持することができます。

Step 1: 4D v11 SQLの2つのインスタンスの起動

同時に4D v11 SQLの2つのインスタンスを起動させるには、2番目の実行可能なファイルが存在しなければなりません。

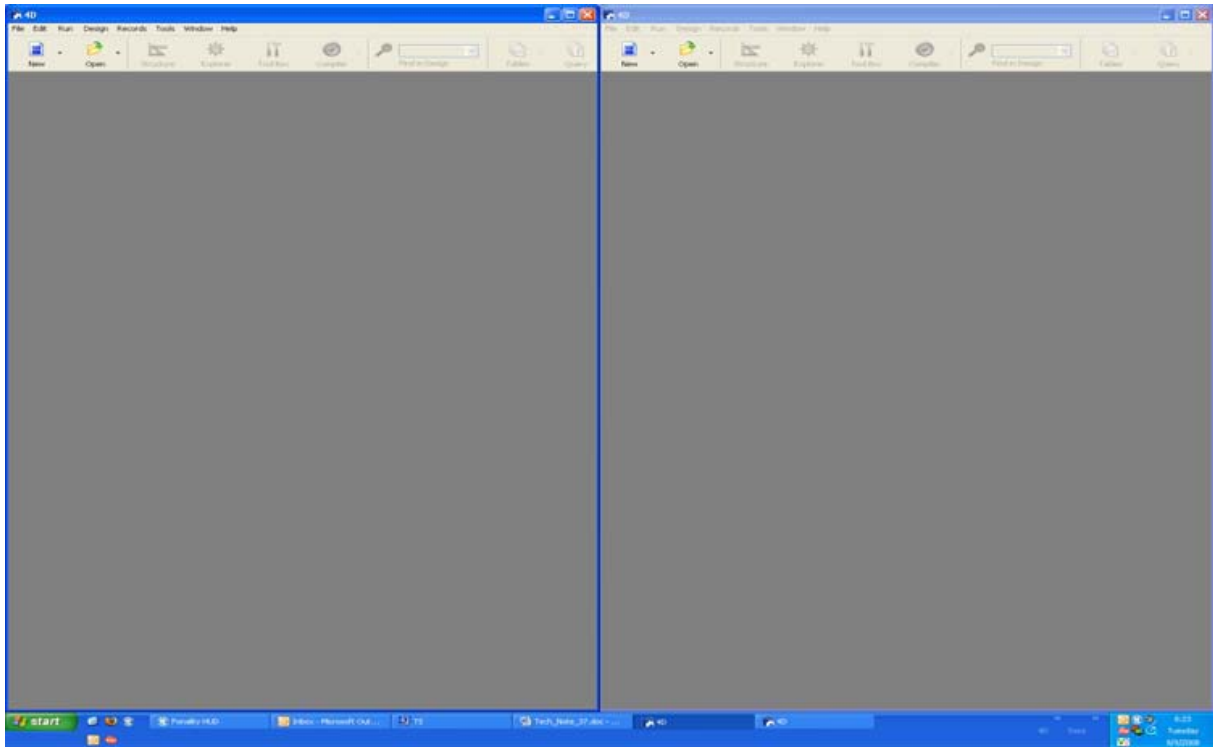
これを行うには新しいフォルダを作成し、4D v11 SQLアプリケーションフォルダのコンテンツをコピー&ペーストするだけです。

Windowsでは、4D v11 SQL アプリケーションのデフォルトパスは以下のとおりです。

C:\Program Files\4D\

Macのデフォルトパスは以下のとおりです。

Macintosh HD:Applications:4D:



Two instances of 4D v11 SQL running simultaneously.

Step 2: ごみ箱の中のテーブルの処理

ごみ箱の取り扱い方は、オリジナルデータベースのデザイン/コーディング方法により、異なります。

ハードコードされたテーブル番号（例：**ALL RECORDS(Table(2)->)**）の参照がオリジナルデータベースにある場合は、ごみ箱の中のすべてのテーブルはデータベースストラクチャに戻さなければなりません。その理由は次のステップに関連します。次のステップでは、データストラクチャのXMLファイルへの書き出しを実行するのですが、ストラクチャがXMLファイルへ書き出された時、もしテーブルが取り除かれていた場合は、テーブルに新しい参照番号が割り当てられます。例えば、以下のテーブルが作成されたとします。

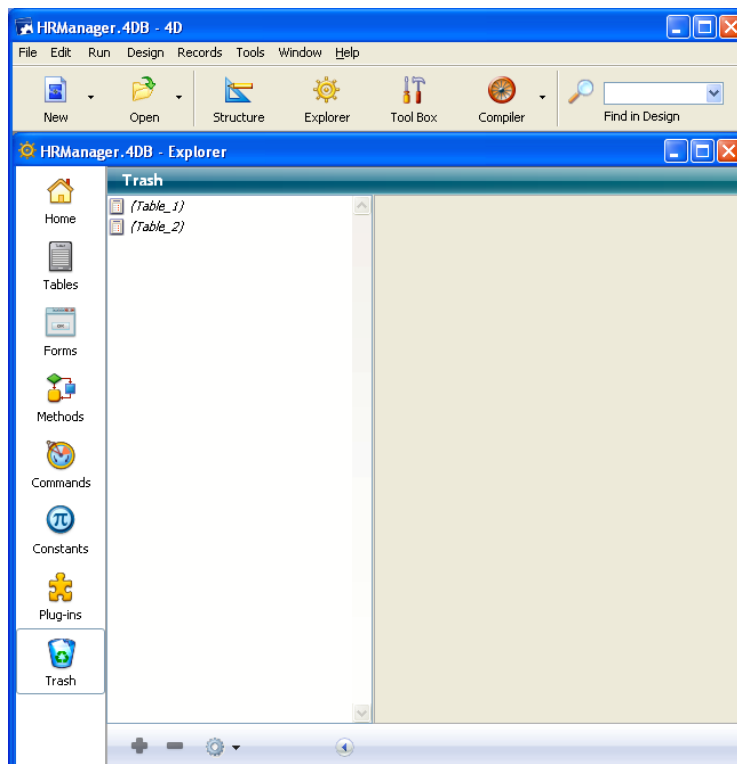
テーブル名	テーブル番号
Table_1	1
Table_2	2
Table_3	3
Table_4	4
Table_5	5

そして、その後Table_2とTable_4は削除されます。XMLファイルのストラクチャを使用して、ストラクチャが書き出され、新しいデータベースが作成されます。その際、新しい参照番号は以下のようになります。

テーブル名	テーブル番号
Table_1	1
Table_3	2
Table_5	3

上記のように、テーブル番号はいつも同じではありません。そして（ハードコードされた）テーブル番号の参照により、エラーが発生することもあります。

テーブル番号にハードコードされた参照のないテーブルがごみ箱にある場合、ごみ箱を空の状態にします（つまり、ごみ箱にはテーブルが存在しない）。このステップを怠ると次のステップでエラーが発生します。



There should be no tables in the trash.

Step 3: データベースストラクチャの読み込み／書き込み

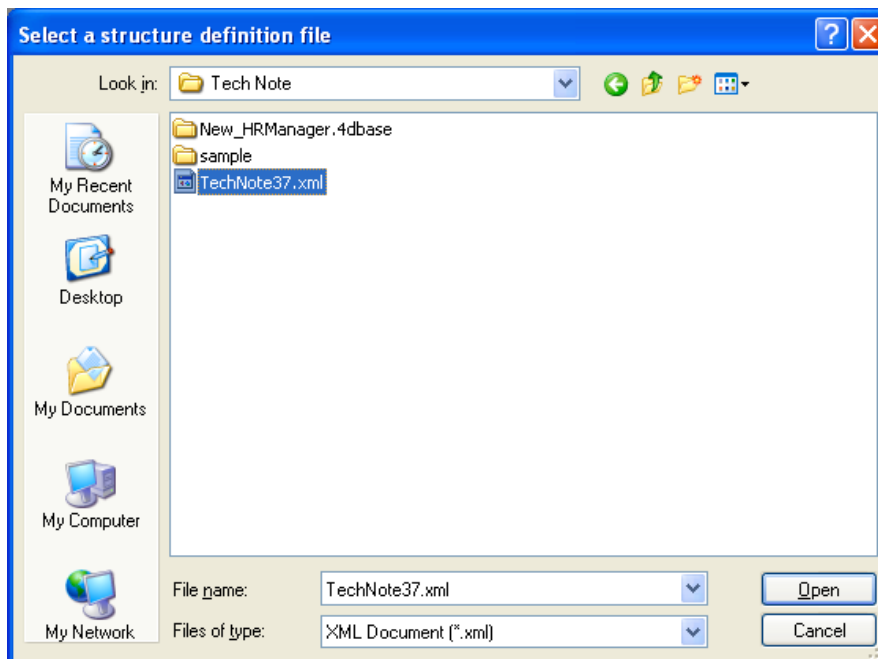
ごみ箱の中のテーブルを処理した後、データベースストラクチャの書き出しを実行します。デザインモードのメニューバーで以下のとおりに進みます。

ファイル>書き出し>ストラクチャ定義をXMLファイルに…

ストラクチャファイルを保存する任意の場所を選び、保存します。次のステップで作成したばかりのストラクチャファイルを使用して、新しいデータベースを作成します。2番目の4D v11 SQLインスタンスのメニューバーで以下のとおりに進みます。

ファイル>新規>ストラクチャ定義から新規データベース…

オリジナルデータベースと同じストラクチャを持つ新しいデータベースが作成されました。



Select the XML file containing the structure of the database.

新しいデータベースで、単にテーブルをドラッグ&ドロップしただけでは、関係とテーブル番号は保持されませんので注意して下さい。

XMLによるストラクチャファイルの読み込み／書き出し方法に関する更に詳しい情報は、*4D v11 SQL*のテクニカルノート、08-26, *XML Structure Import/Export*を参照して下さい。

リソースファイル (変換データベース)

オリジナルデータベースが4Dの以前のバージョンから4D v11 SQLへ変換されている場合、必ずこのステップを実行して下さい。データベースが4D v11 SQLでデザインされている場合、Step 5へ進んで下さい。

4D 2004以前のデータベースでは、参照用にリソースファイル(.RSR)を使用します。4D v11 SQLではリソースファイルは使用しませんが、データベースが4D 2004 (またはそれ以前)から4D v11 SQLへ変換されている場合は**.RSR** ファイルを使用します。

この場合、新しいデータベースで参照をそのままの状態に保つため、**.RSR**ファイルを必要とします。

真新しいデータベースを作成しているため、**.RSR**ファイルは自動的に読み込まれません。そのため手作業で読み込まなければなりません。

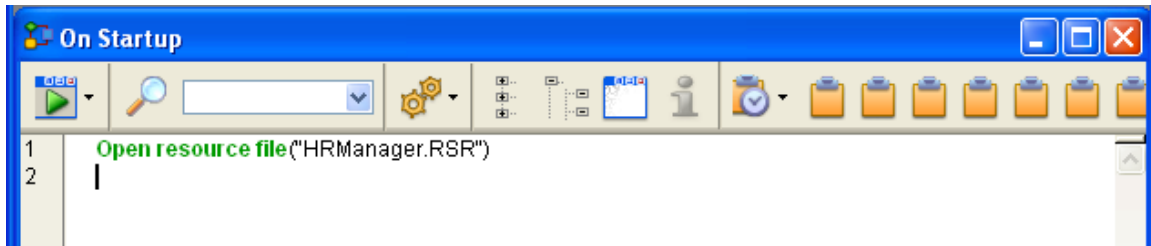
オリジナルのデータベースフォルダで、**.RSR**ファイルをコピーし新しいデータベースのフォルダにペーストします。

*On Startup*データベースメソッドの1行目に以下を追加します。

```
Open Resource File ("myDatabase.RSR")
```

myDatabase.RSRは、今コピーしたファイルの名前です。

Open Resource File コマンドは、参照が保持されていることを確認し、**.RSR**を開きます。



The resource file (.RSR) must be opened manually in newly created databases.

Step 5: ツールボックス

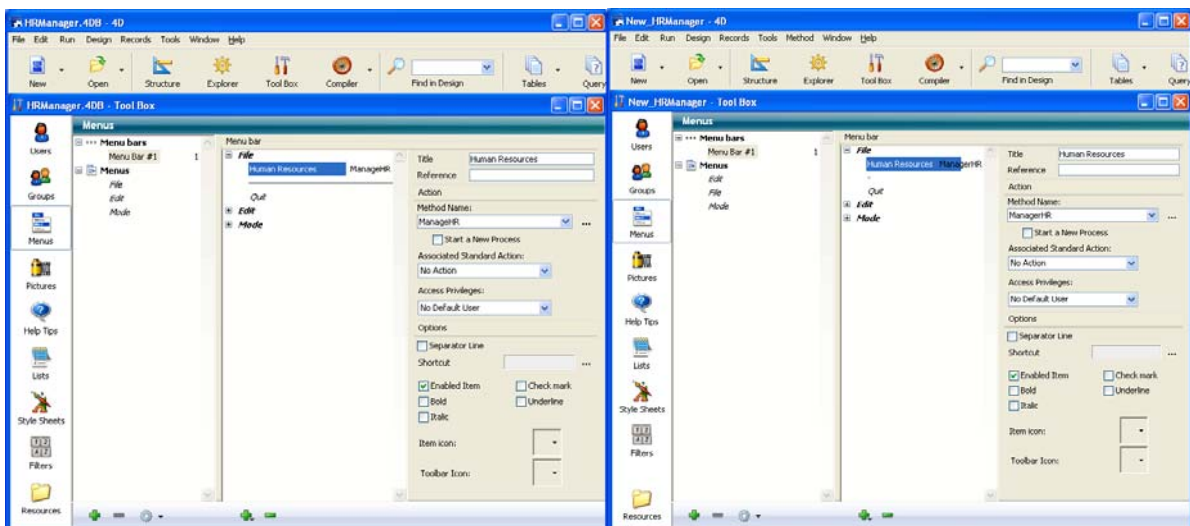
ツールボックスにあるオブジェクトを移動させます。

ユーザとグループ

ユーザとグループは新しいデータベースへ移すことはできませんので、手作業で作成して下さい。

メニュー

メニューは、新しいデータベースへドラッグすることはできませんので、手作業で作成しなければなりません。同じ参照番号を保持しながら、オリジナルデータベースを作成した時と同じ手順で作成するようにして下さい。デフォルトメニューバー (“Menu Bar #1”) については、オリジナルデータベースの最初のメニューバーと同じになるように編集して下さい。



Menus must be created manually.

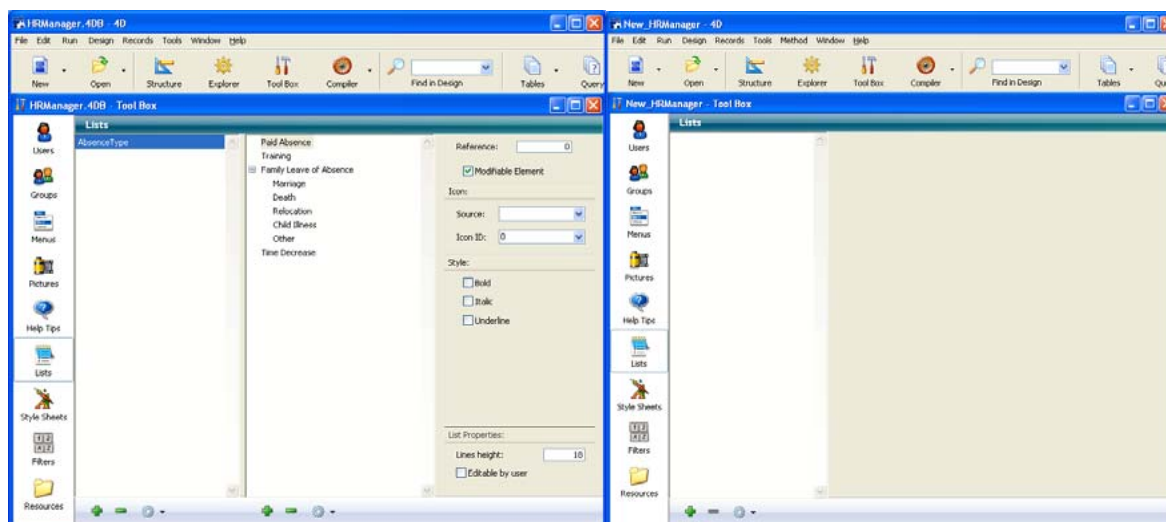
ピクチャ/Help Tips/リスト/フィルタ/リソース

ピクチャ、ヘルプリスト、リスト、フィルタとリソースは、ドラッグ&ドロップで、新しいデータベースに追加できます。同じウィンドウで両方のデータベースのツールボックスを開き、オブジェクトを古いデータベースから新しいデータベースへドラッグするだけです。

注：これらのオブジェクトを一括で移すことはできません。1個単位で移します。

スタイルシート

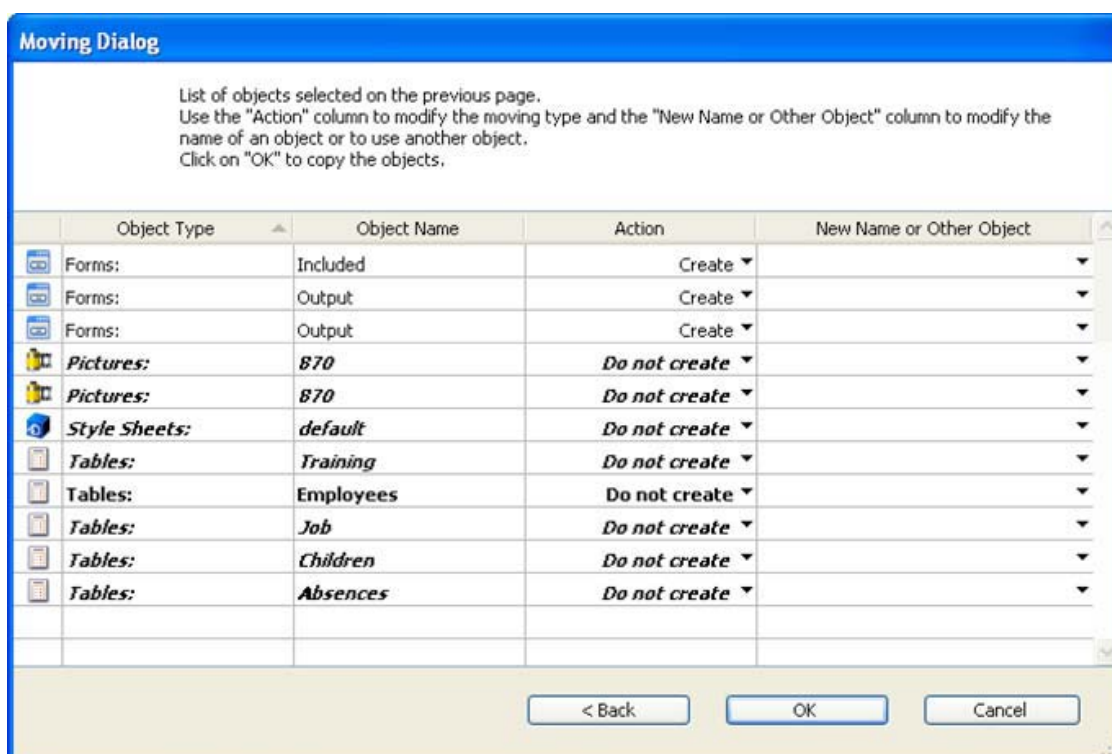
上記で説明したように、スタイルシートについてはドラッグ&ドロップしなければなりません。しかし、デフォルトスタイルシートは例外です。新しいデータベースヘドラッグせず、オリジナルのデータベースと同様、新しいデータベースにデフォルトスタイルシートを設定します。



Pictures, Help Lists, Lists, Filters, and Resources can be dragged to the new database.

次に進む前の重要な注意

この時点から、オブジェクトを移した時、そのオブジェクトをどのように処理するかを促すダイアログが表示されます。例えば、フォームを移したとき、そのフォームで移すと参照されたすべてのオブジェクトが表示され、それらのオブジェクトに対しての処理法がリクエストされます。通常のルールとして、新しいデータベースで既に作成された各オブジェクトに関しては、Moving Dialogウィンドウで“Do not create”をその処理として選びます。

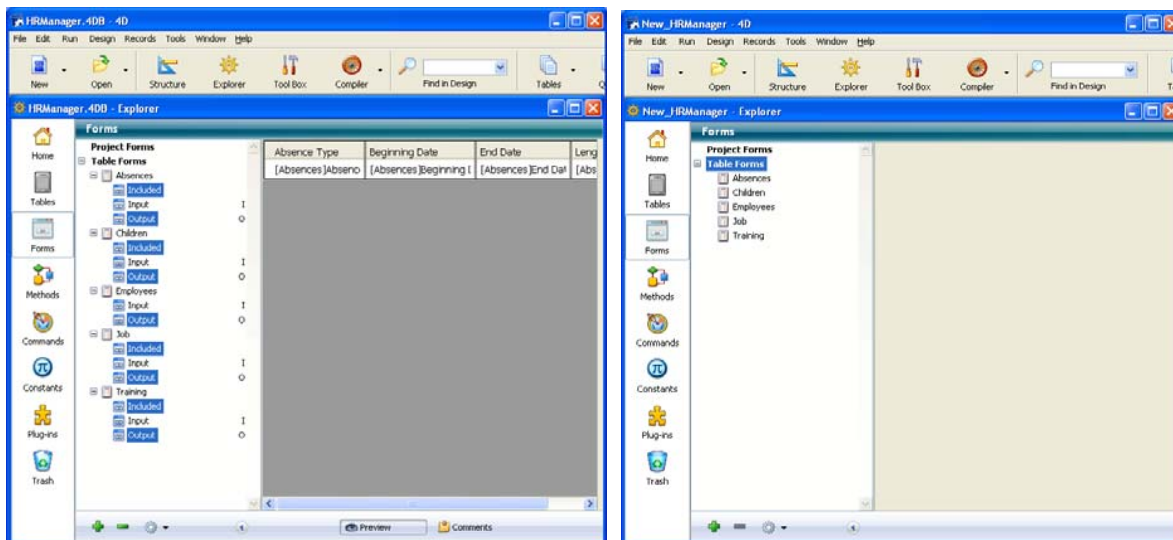


If the object is already present in the new database, do not create it again.

Step 6: フォームの移動

リストフォーム

オリジナルデータベースのエクスプローラウィンドウで、すべてのリストフォーム（CTRLを押しながら任意のアイテムをクリック）を選択し、新しいデータベースへドラッグします。



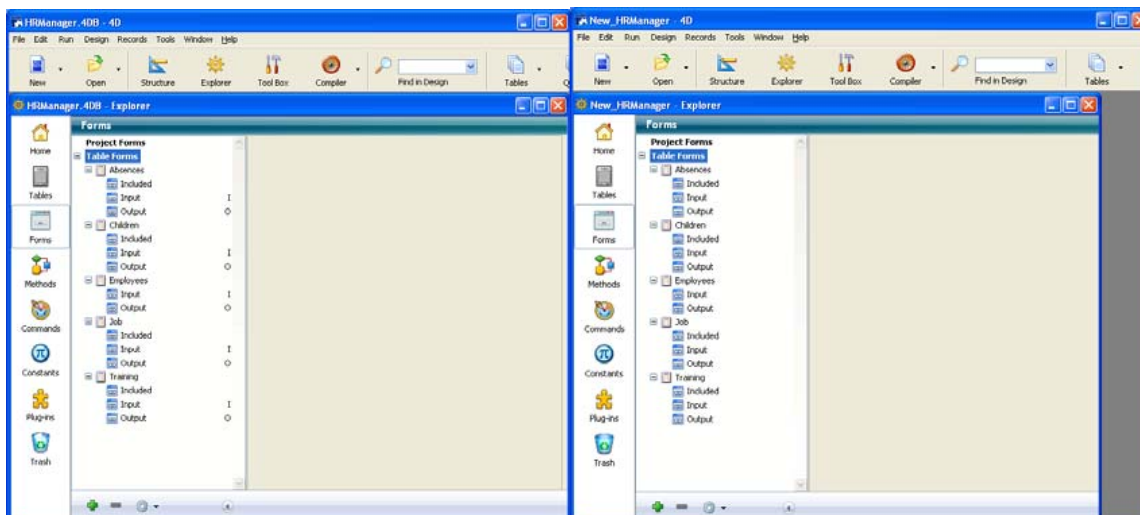
First move the List Forms to the new database.

詳細フォーム

そして、詳細フォームを選択し、新たに任意の場所へドラッグします。

入出力フォームの設定

現在、オリジナルデータベースからドラッグしてきたすべてのフォームが、新しいデータベースにもあります。しかし、“入力”または“出力”のいずれかのフォームとして、これらのフォームには割り当てられていません。オリジナルデータベースと同様に、新しいデータベースの各フォームに“入力”または“出力”を割り当てます。



Input / output forms must be assigned manually.

Step 7: メソッド

プロジェクトメソッド

エクスプローラウィンドウのメソッドで、すべてのメソッドを選択し、新しいデータベースへドラッグします。

データベース／トリガメソッド

データベースとトリガメソッドを、新しいデータベースへドラッグすることはできません。新しいデータベースの各メソッドでそれぞれのメソッドのコードをコピー&ペーストして下さい。

Step 8: プラグイン

プラグインをインストールするには、アプリケーションの *plugins* フォルダへ必要なプラグインの *PluginName.bundle* フォルダをコピーして下さい（またはプラグインファイルを古い形式の プラグイン用の Win4DX/Mac4DX フォルダへ移す）。

Conclusion

このテクニカルノートでは、オブジェクトのドラッグ&ドロップによるデータベースの複製方法を説明しました。また、新しいデータベースでその機能を維持するため、ステップの順序の重要性を記述しました。

ドラッグ&ドロップ機能により、デベロッパは古いデータベースから新しいデータベースを容易に作成することができ、仕事量と時間の節約にもなります。

Related Resources

4D v11 SQL のテクニカルノート 08-26, *XML Structure Import/Export*