

# Custom Users and Groups Import/Export

By Josh Fletcher, Technical Support Engineer, 4D Inc.  
Technical Note 07-12

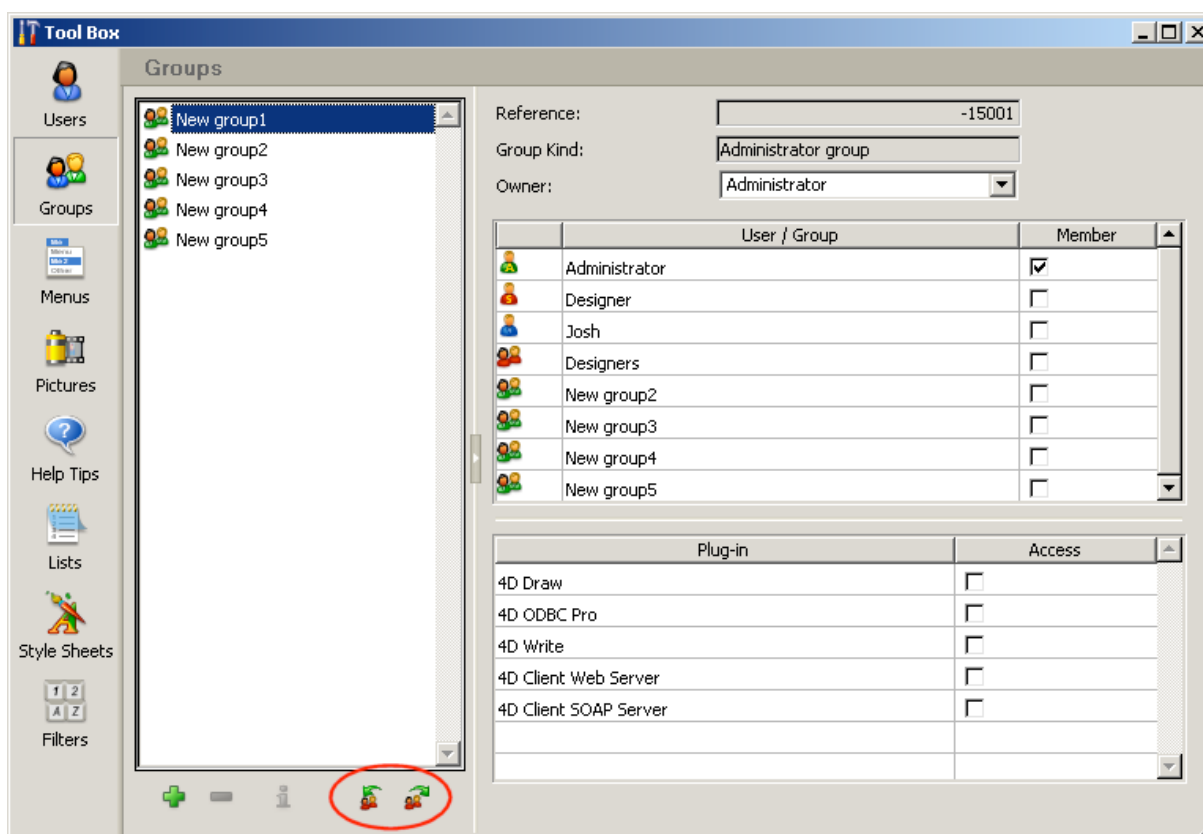
## Abstract

このテクニカルノートではUsers and Groupsのカスタマイズされた書き出し/読み込み実装の例題を提供します。特に4Dではサポートされていないプラグインアクセス権の書き出し/読み込み機能を実装している点が注目できます。

デモデータベースにはソースコードが含まれています。

## Introduction

4D 2004 ではAdministratorによって作成されたユーザとグループを書き出し、また読み込むことができます。この機能はツールボックスのグループページにあります:



デザインモードのほかに、**EDIT ACCESS**コマンドを使用してこの機能にアクセスすることもできます。

4D 2004 Design Referenceより:

管理者はグループを保存することができるため、データベースのアクセスシステムを保存して、そのシステムを同じデータベースの修正版や新規データベースに移行することができます。これは、新しいバージョンのデータベース用にアクセスシステムを再構築する上で非常に役立ちます。グループ

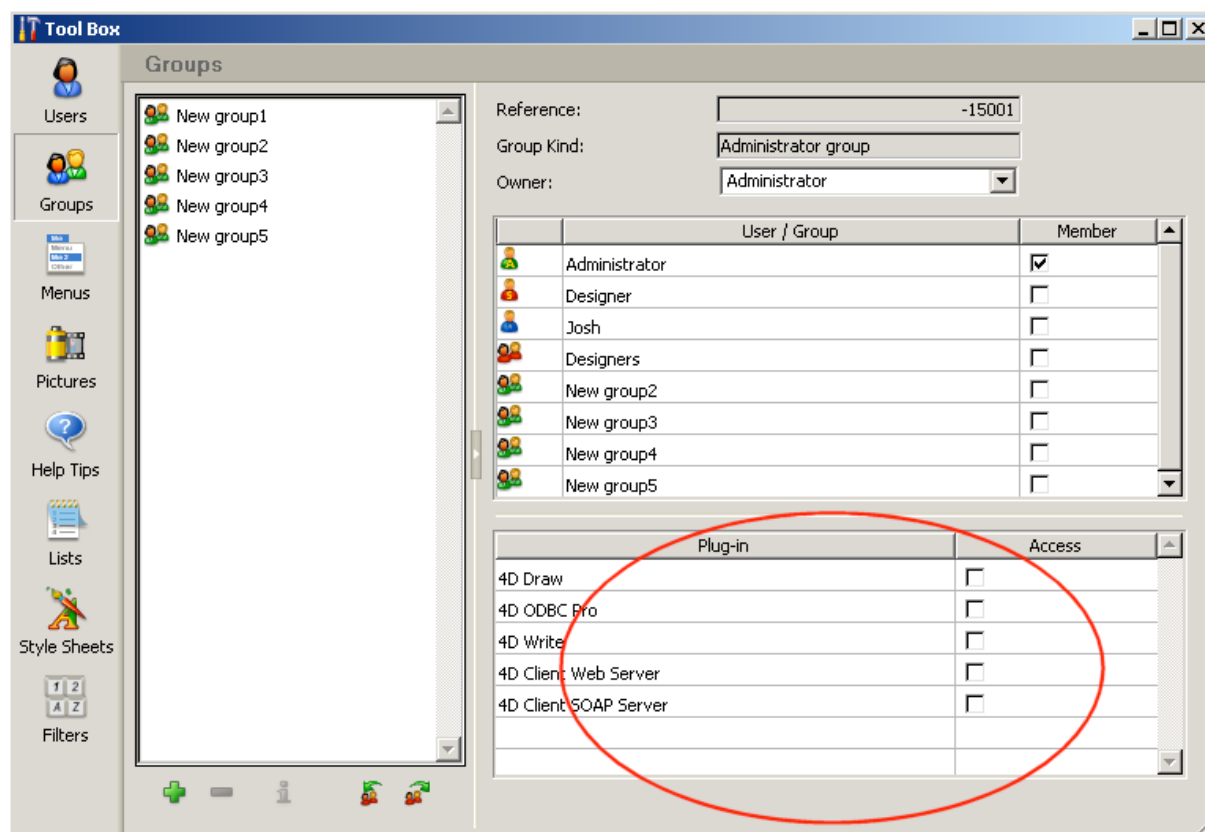
を再度読み込むことができるため、データベースのユーザは新しいアクセスシステムを覚える必要がなくなります。

ユーザ名、パスワード、*Startup* メソッド名、グループ、グループオーナー、グループのメンバーはすべて保存されます。

しかしこの機能は、プラグインアクセス権の書き出しや読み込みは行いません。異なるデータベースにユーザ&グループ設定を読み込む際は、プラグインアクセス権を手動で設定しなければなりません。これは多数のグループやプラグインがインストールされた環境では大変な作業になります。

## 4D Plug-In Access Settings

4Dでは特定のグループにプラグインのアクセス権を与えることができます。この設定は以下のようにツールボックスで行われます：



プラグインアクセス権を設定することで、開発者や管理者は特定のプラグインを使用することのできる人数を制限することができます。これはセキュリティ上の理由で、またはライセンス数の制限のために行われます（クライアントサーバシステムで重要です）。

これらの設定の書き出し/読み込みはユーザ&グループの書き出し/読み込み機能のカスタマイズの一例です。

## Customization – Step One – Users and Groups

デザインモードへのアクセスは制限されていたり、コンパイルモードのようにまったく行え

ない場合もあります。そのためツールボックスはユーザ&グループの読み込みや書き出しのために常に使用できるわけではありません。もちろん**EDIT ACCESS**コマンドを使用して、ツールボックスをメニューから表示させることもできます。しかしこの場合でも多くはカスタマイズできません。

カスタマイズプロセスの第一歩は、プログラマ的にユーザ&グループを取得することです。これは4Dコマンドの**USERS TO BLOB** や **BLOB TO USERS**を使用します。

## USERS TO BLOB

4D Language Reference には **USERS TO BLOB** について以下のように書かれています：

**USERS TO BLOB** コマンドは、すべてのユーザアカウントのリストと、*Administrator* (管理者)が作成したデータベースグループを *BLOB* タイプの引数<ユーザ>に格納します。

生成された *BLOB* は自動的に符号化され、**BLOB TO USERS** コマンドを使用した場合にのみ読み取ることができます。この *BLOB* は、ハードディスク上のファイルやフィールドに保存可能です。

このコマンドを使用して、ユーザ&グループを簡単に書き出して、また4DのBLOB操作コマンドを使用して適当な場所に保存することができます。

## BLOB TO USERS

4D Language Reference には **BLOB TO USERS** について以下のように書かれています：

**BLOB TO USERS** コマンドは、*BLOB* タイプの引数<ユーザ>に格納されたユーザアカウントをデータベースへ追加します。*BLOB*<ユーザ>は符号化されており、**USERS TO BLOB** コマンドを用いて作成されていなければなりません。

**USERS TO BLOB, BLOB TO USERS** により、正しく構築されたBLOBを用意することで、いかなるソースからもユーザ&グループが読み込めることになります。

## Step One Complete

プログラマ的にユーザ&グループにアクセスすることのできる方法により、書き出し/読み込みがカスタマイズできるようになります。さらに情報を適当な方法で格納することができます。

このテクニカルノートに含まれる例題コードでは、XMLファイルに情報を格納します。

また "ユーザ&グループ"テーマ内の4Dコマンドを使用して書き出し/読み込みをカスタマイズできる点にも注目してください。

このテクニカルノートではプラグインアクセス権に関するコマンドを見ることにします。

## Customization – Step Two – Plug-In Access

---

プラグインアクセス権を書き出し/読み込みするには、4Dの二つのコマンド、**SET PLUGIN ACCESS** と **Get plugin access**を使用します。

## SET PLUGIN ACCESS

**SET PLUGIN ACCESS** コマンドは、データベースにインストールされた「シリアルライズされた」プラグインの使用をグループに対してプログラムで許可します。これによりどのようにプラグインライセンスを使用するか管理することができます。

シリアルライズされたプラグインのリストは4Dランゲージリファレンスにあります。

プラグインに対し一度に一つのグループを割り当てることができることに注意してください。このコマンドの実行に成功すると、他のグループが既に持っていたそのプラグインへのアクセス権を失います。

またこのコマンドはインストールされ、ライセンスを持ち、有効になっているプラグインのアクセス権のみを変更することに注意してください。この条件に当てはまらない場合、プラグインアクセス権は設定されず、OKシステム変数は0に設定されます。

## Get plugin access

**Get plugin access** コマンドは、シリアルライズされたプラグインに対し利用を認証されたグループ名を返します。プラグインにグループが割り当てられていなければ、このコマンドは空の文字列を返します。

シリアルライズされたプラグインのリストは4Dランゲージリファレンスに記載されています。

**Get plugin access**はプラグインがインストールされ、ライセンスを持ち、有効になっている場合のみグループ名を返すことに注意してください。これらの条件に当てはまらない場合、グループ名は返されません。

## What's next?

プラグインアクセス権を含むユーザ&グループの読み込み/書き出しモジュールをカスタマイズするために必要なコマンドを説明しました。例題データベースに実装された内容については“The Source Database”の節を参照してください。

## Customization – Step Three, and Beyond...

---

このテクニカルノート of 例題データベースでは、プラグインアクセス権の書き出し/読み込みのみを扱っています。ユーザ&グループテーマには多数のコマンドが用意され、さらなるカスタマイズが可能となっています。例えば、グループのメンバを**Set user properties**を使用して変更することが可能です。このテーマのコマンドを見て、可能性を探ることをお勧めします。詳細は4Dランゲージリファレンスを参照してください。

## Implementation - The Source Database

---

この節では、例題データベースに実装されたユーザ&グループの書き出し/読み込みモジュールについて説明します。

## Overview

このテクニカルノートに含まれる例題データベース (“UG\_Source.4DB”) には、ユーザ&グループの書き出し/読み込みモジュールを実装していて、プラグインアクセス権を含むユーザ&グループの情報を書き出したり読み込んだりすることができます。

情報は4DのDOMコマンドを使用してXMLフォーマットで保存されます。詳しく説明すると、4Dのユーザ&グループはBase64で符号化されたBLOBとして書き出され、プラグインアクセス権はXMLとして書き出されます。

データベースには約40のプロジェクトメソッドが含まれています。しかしそのほとんどはインターフェースを表示するために使用されています。ユーザ&グループの書き出し/読み込みのカスタマイズを理解するために見る必要のあるコードは4つです:

- *UG\_EXPORT\_UGToXMLTree* – ユーザ&グループをXML形式で保存します。
- *UG\_EXPORT\_PluginAccessToXMLTree* – プラグインアクセス権をXML形式で保存します。
- *UG\_IMPORT\_XMLTreeToUG* - ユーザ&グループをXMLから読み込みます。
- *UG\_IMPORT\_XMLTreeToPluginAccess* - プラグインアクセス権をXML から読み込みます。

これらのメソッドは以下で説明します。

最後に、すべてのメソッドはコメントで説明されています。

## *UG\_EXPORT\_UGToXMLTree*

このメソッドはシンプルで、**USERS TO BLOB**を使用するサブルーチンをコールし、結果のBLOBを**ENCODE**コマンドでBase64にエンコードします。そしてBLOBを渡されたXMLに格納します:

```
$encodedUsers:=UG_EXPORT_CreateEncodedUGBlob  
$elemRef:=DOM Create XML element($rootRef;<>UG_XPATH_USERGROUPSBLOB)  
DOM SET XML ELEMENT VALUE($elemRef;$encodedUsers)
```

このBLOBを格納する、<>UG\_XPATH\_USERGROUPSBLOBに割り当てられたXPathは*UG\_Startup*メソッドで定義されます。

以下は*UG\_EXPORT\_CreateEncodedUGBlob* サブルーチンの一部です:

```
USERS TO BLOB($users)  
ENCODE($users)  
$0:=$users
```

**ENCODE** コマンドを使用して、BLOBを確実にXMLに格納できるようにします。XMLのCDATAセクションを同じ目的で 사용할 ことを覚えておいてください。

## *UG\_EXPORT\_PluginAccessToXMLTree*

このメソッドはプラグインアクセス権を渡されたXMLに格納します。設定は**Get plugin access**コマンドを使用して取り出されます。

重要: **Get plugin access** はプラグインがインストールされ、ライセンスを持ち、有効になっているときにのみ有効です。

このメソッドは**UG\_Startup**で定義されるインタープロセス変数をいくつか使用します。詳細は当該メソッドの節を参照してください。

```
For ($i;1;<>UG_TOTALPLUGINS)
    $groupWithAccess:=Get plugin access(<>UG_PLUGINIDS{$i})

    If (OK#1)
        $groupWithAccess:=""
    End if

    If ($groupWithAccess#"")
        $elemRef:=DOM Create XML element($rootRef;<>UG_XPATHS{$i})
        DOM SET XML ELEMENT VALUE($elemRef;$groupWithAccess)
        $anyExported:=True
    End if
End for
```

ループを使用してシリアルライズされたプラグインを処理しています。ループで使用される配列は**UG\_Startup**メソッドで定義されています。

このメソッドはプラグインアクセス権が書き出されなかった場合に警告メッセージを返します。これはプラグインアクセス権を書き出す必要がなければ、このメソッドはコールされるべきではなく、もしプラグインアクセス権が書きだされなければプラグインに何かおかしい点があるということに基づいています。これは技術的にはエラーではないのでエラーメッセージは返しません。プラグインアクセス権が書き出された場合はエラーメッセージも警告メッセージも返されません。

## **UG\_IMPORT\_XMLTreeToUG**

このメソッドはサブルーチンをコールします。サブルーチンは4DのDOMコマンドを使用して、XMLからエンコードされたBLOBを取り出し、**DECODE**コマンドを使用してでコードを行います。そして**BLOB TO USERS**コマンドでユーザ&グループ設定を読み込みます:

```
$result:=UG_IMPORT_GetUGBlob ($rootRef;->$users)
If ($result=UGM_SUCCESS )
    BLOB TO USERS($users)
End if
```

以下は **UG\_IMPORT\_GetUGBlob** サブルーチンの一部です:

```
$elemRef:=DOM Find XML element($rootRef;<>UG_XPATH_USERGROUPSBLOB)
If (OK=1)
    DOM GET XML ELEMENT VALUE($elemRef; $pUsers->)
    If (OK=1)
        DECODE($pUsers->)
    End if
End if
```

<>UG\_XPATH\_USERGROUPSBLOB に割り当てられたBLOB要素のXPathは**UG\_Startup**で定義

されています。

## **UG\_IMPORT\_XMLTreeToPluginAccess**

このメソッドはプラグインアクセス権をXMLから取り出し、データベースに読み込みます:

```
For ($i;1;<>UG_TOTALPLUGINS)
  $elemRef:=DOM Find XML element($rootRef;<>UG_XPATHS{$i})
  If (OK=1)
    DOM GET XML ELEMENT VALUE($elemRef;$groupWithAccess)
    If (OK=1)
      If ($groupWithAccess#"")
        SET PLUGIN ACCESS(<>UG_PLUGINIDS{$i};$groupWithAccess)
      End if
    End if
  End if
End for
```

シリアルイズされたプラグインを処理するためにループが使用されます。ループで 사용되는配列は**UG\_Startup**メソッドで定義されています。

**Note:** このコマンドは少なくとも一つのプラグインアクセス権設定が見つかり設定されれば成功したものとみなします。**SET PLUGIN ACCESS**のコールに失敗すれば完全に失敗したとみなします。プラグインアクセスが設定されない場合警告が返されます。

## **UG\_Startup**

コードをきれいにするために、いくつかの配列がこのメソッドで定義され、シリアルイズされたプラグインIDやプラグイン情報を記述するXPATHを格納しています。

つまり以下のようなコードブロックをプラグイン事に記述する代わりに、配列を使用しているということです:

```
$groupWithAccess:=Get plugin access( <plug-in ID constant> )
If (OK#1)
  $groupWithAccess:=""
End if

If ($groupWithAccess#"")
  $elemRef:=DOM Create XML element($rootRef; <XPATH for this plug-in> )
  DOM SET XML ELEMENT VALUE($elemRef;$groupWithAccess)
  $anyExported:=True
End if
```

配列を使用することで、forループを使用することができます。またコードで “magic numbers” の使用を避けることができますようになります。

## **Limitations and Enhancements**

この節では現在の書き出し/読み込みモジュールの制限と拡張について考察します。

- ・4Dのユーザ&グループはBLOBで書き出されるので、表示や人間が読むことのできる表現をXMLから取り込むことはできません。XMLを開くと、ユーザ&グループはBase64でエンコードされたテキストブロックが表示されます。



もしユーザやグループの実際の名前をXMLに表示する必要がある場合は、4Dランゲージのユーザ & グループテーマの他のコマンドを使用して行うことができます。

- 書き出しと読み込みはそれぞれ別のフォームに実装されています。必要であれば簡単にそれを一つのフォームに統合することができます。

- XML要素のXPATHは、データベースのスタートアップコードの中にハードコードされています。このおかげでループの中で設定を書き出し/読み込みできるようになっています。

同様にシリアルライズされたプラグインIDもハードコードされています。一つの拡張のアイデアはXMLファイルのDTDを定義し、シリアルライズされたプラグインとそのIDを追加したり削除したりできるようにします。この方法を使用すればカスタマイズされた書き出し/読み込みモジュールがより動的で堅牢になります。

## Other Design Decisions

この情報はテクニカルノートの特ピックにとって重要というわけではありませんが、例題データベースをより深く理解していただくために記述されます。

このデータベースは番号ベースの返り値ではなく、メッセージングシステムを採用しています。それぞれのメソッドはエラーコードを返す代わりにメッセージ文字列を返します。メッセージは定数として扱われ、特定のプロジェクトメソッドが定数として使用されます。この方法を採用したのは以下の理由からです：

- エラーコードを抽象化すると、コード中の“magic numbers”を取り除くことができます。この方法では、コンパイラはスペル間違いをキャッチすることができます。対して文字列や数値リテラルを使用すると、コンパイラはそれを無視します。またメソッドエディタのコードフォーマットはスペル間違いのメソッド名を指摘することができます。

- 独自の定数の実装は、4Dの機能としてドキュメント化されていません。メソッドベースのシステムは、将来定数に関する4Dの実装が変更された場合も確実に動作します。

- 4Dのコード入力補完機能を、定数のそれと同じように使用することができます。

データベースでは以下の命名規則を使用しています：

**UGM\_** このメソッドはエラーレポートに使用されます。

**UGM\_EX\_** 書き出しに関するメッセージ。

**UGM\_IM\_** 読み込みに関するメッセージ。

**UG\_UTIL\_** このテクニカルノートに直接関係のないヘルパメソッド。

**UG\_EXPORT\_** 書き出しコード。

**UG\_IMPORT\_** 読み込みコード。

**UG\_M\_** メニュー項目から呼ばれるメソッド。

**UG\_P\_** プロセスで使用されるメソッド。

**UG\_** 他のメソッド

## The Demo Database

---

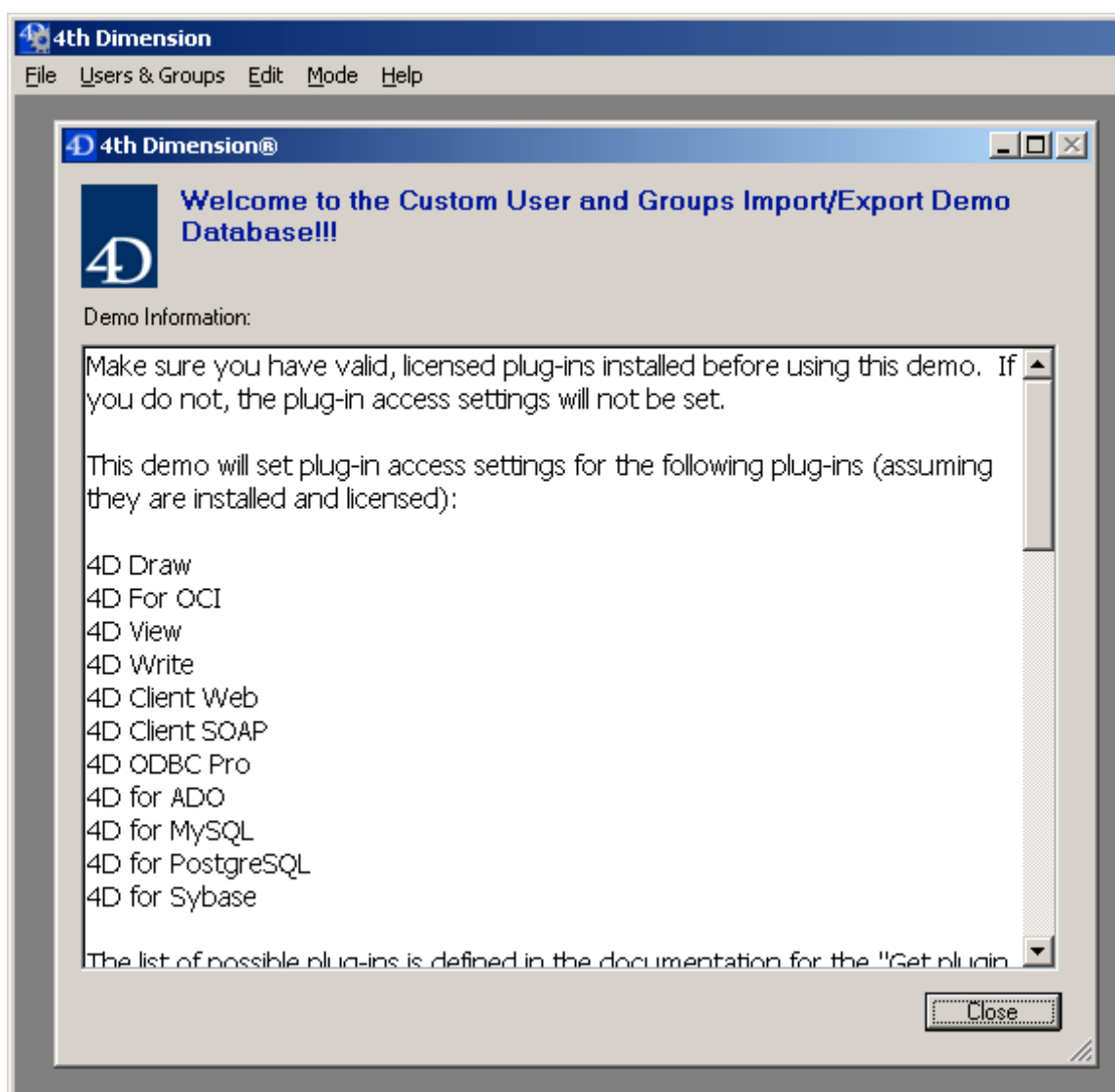
このテクニカルノートに含まれるでもデータベースは、基本的にソースデータベースと同一



です。しかしユーザへの案内コードとサンプル書き出しファイルが含まれています。サンプルの書き出しファイルには、いくつかの4Dユーザとシリアルライズされたプラグイン毎のグループ、そしてプラグインアクセス設定が含まれています。

例題の書き出しファイルには、シリアルライズされたすべてのプラグインのアクセス権設定が含まれていますが、プラグインの設定により、すべてが読み込まれるわけではないことに注意してください。

でもデータベースを使用するには、“UG\_Demo.4DB”を4Dで起動します:



デモを使用するには:

デモを使用する前に有効なプラグインライセンスを所有し、それらがインストールされていることを確認してください。そうでなければプラグインアクセス権は読み込まれません。

このデモでは以下のプラグインのアクセス権を設定します:

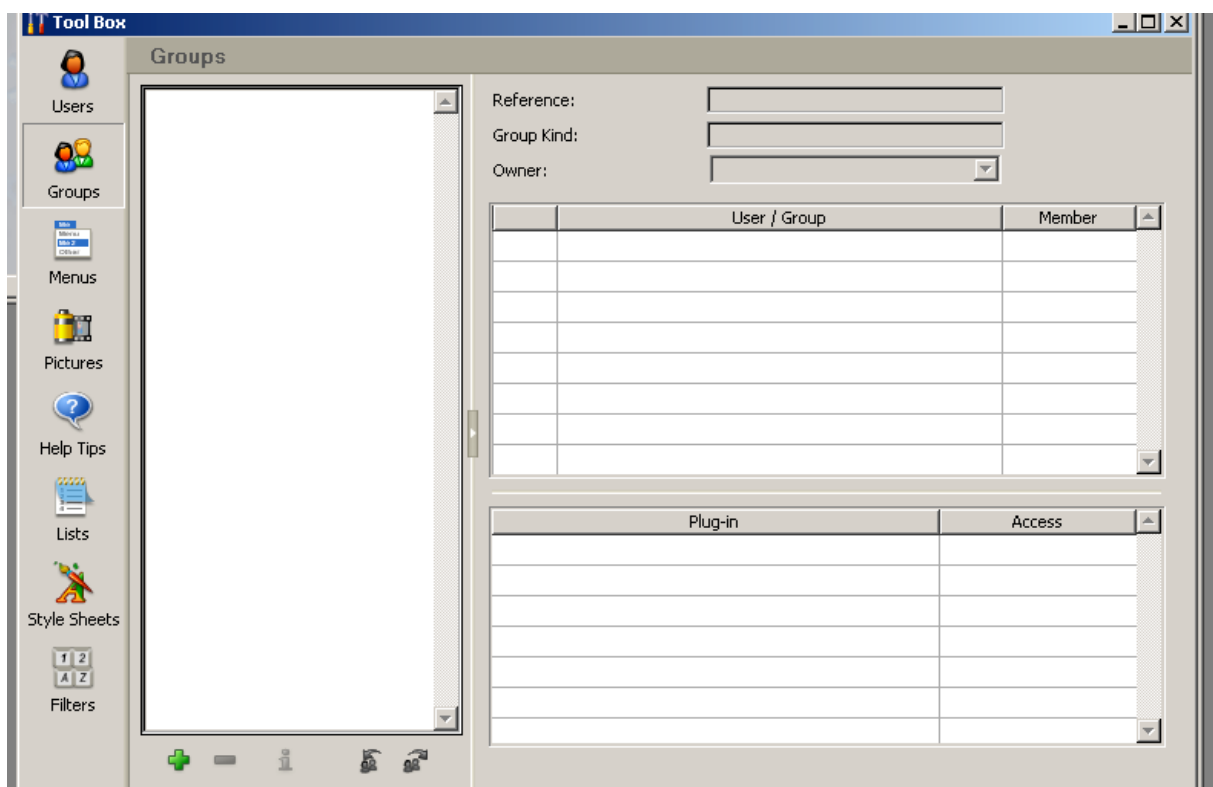
- 4D Draw
- 4D For OCI
- 4D View
- 4D Write

- 4D Client Web
- 4D Client SOAP
- 4D ODBC Pro
- 4D for ADO
- 4D for MySQL
- 4D for PostgreSQL
- 4D for Sybase

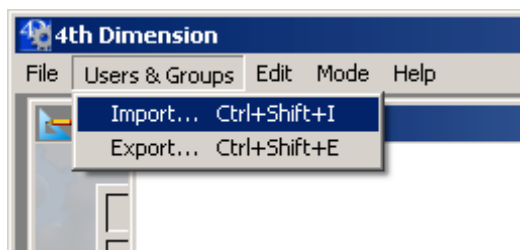
利用可能なプラグインのリストは[Get plugin access](#) コマンドの説明で見ることができます。

デモのステップ:

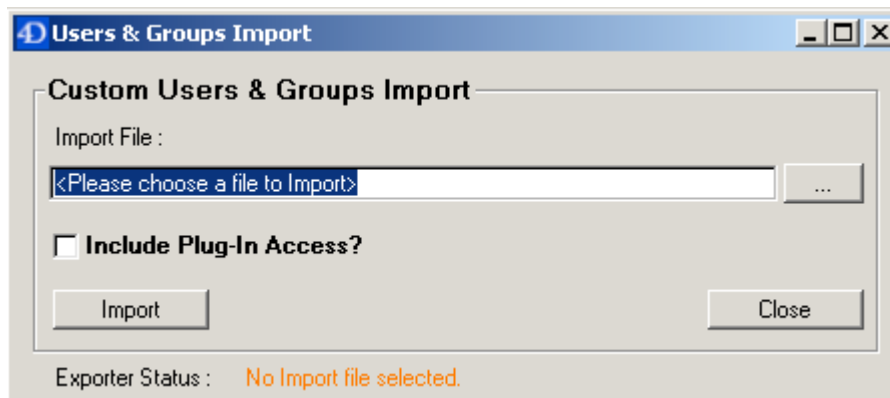
- デザイナーのパスワードを作成し、パスワードアクセスシステムを有効にします。またグループが定義されていないことも確認してください:



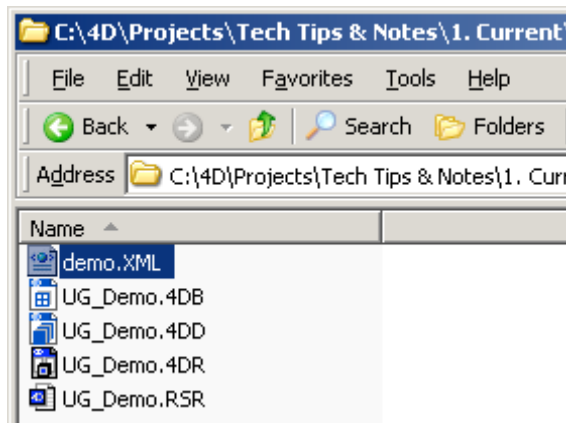
- データベースを再起動します。
- Administrator としてログインします。(パスワードはありません)
- "User & Group"メニューを開き"Import..."を選択します。



- 読み込みダイアログが開きます:

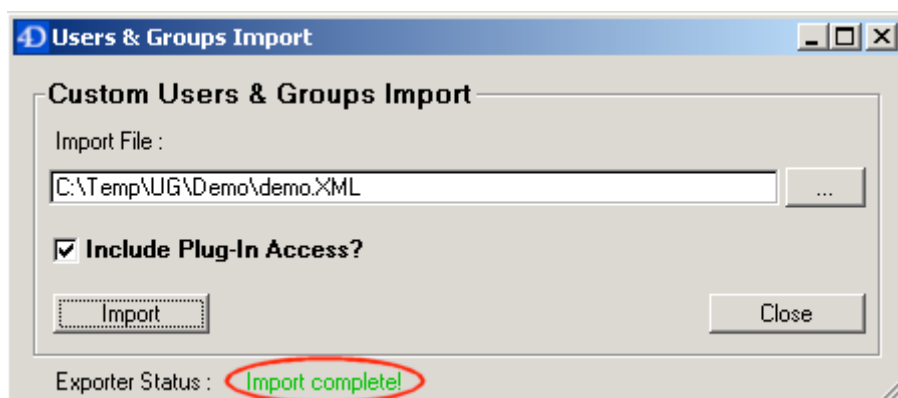


- ストラクチャと同階層にある "demo.XML" を探します。



- "Import Plug-In Access?" チェックボックスをチェックします。
- "Import" ボタンをクリックします。

読み込みが終了すると、"Explorer Status : "ラベルの隣にメッセージが表示されます。メッセージの色は読み込み、または書き込みの結果を表します:



- **Green:** 成功、またはエラーがないメッセージ
- **Orange:** 警告
- **Red:** エラー

これは成功した場合の表示です:

読み込みに成功したらデザインモードのツールボックスを開き、プラグインアクセス権設定が読み込まれていることを確認してください。新しいユーザやグループも追加されているはずです。

もしプラグインアクセス権設定が読み込まれず、警告メッセージが表示された場合は、プラグインがインストールされていないか、無効になっているか、またはライセンスがないということです。読み込みに失敗するとエラーメッセージが表示されます。

## Conclusion

---

このテクニカルノートでは、ユーザ&グループの書き出し/読み込みをカスタマイズする方法について説明しました。このモジュールを使用するとプラグインのアクセス権も書き出し/読み込みすることができます。例題データベースにはソースコードが含まれています。