

SOFTIMAGE|XSI SDK Scripting(VBS)チュートリアル

準備

1. スクリプトに関する情報が以下のサイトから得られます。VBS のマニュアルもこちらから入手できますのでダウンロードしてインストールしておくとい良いでしょう。
<http://www.microsoft.com/japan/developer/scripting/>
2. NetView ページ作成のために HTML エディタ (FrontPage など) があると便利です。
3. スクリプトのグローバル設定を行います。
File – User Preference にて、Scripting/Logging タブをクリックします。
使用するスクリプト言語、ログサイズ、ログファイルを設定します。
本チュートリアルでは、“VB Script Language”を用いますので、Scripting Language を“VB Script Language”に設定して下さい。他はデフォルトで結構です。

オンラインマニュアルの参照方法

Script エディタを開き、コマンドバーのヘルプアイコン(?)をクリックします。Programmer's Reference を参照すると使用可能なコマンド群の一覧がデフォルト HTML ブラウザで表示されます。

Commands

SOFTIMAGE|XSI のスクリプト内で使える XSI ネイティブの command 群

Objects

Methods (関数に相当) や Properties (変数に相当) から構成されるプログラミング構造体

Methods

Properties

Constants

Tutorial 1, Tutorial 2 については、**DKN_MyTools.xsiaddon** をインストールして下さい。

Tutorial 1 : ヒストリからのスクリプトボタンの作成

変形済みの Sphere が Get できるスクリプトボタンを作成してみましょう。

ヒストリからのスクリプトボタンの作成

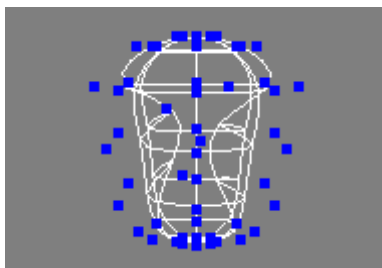
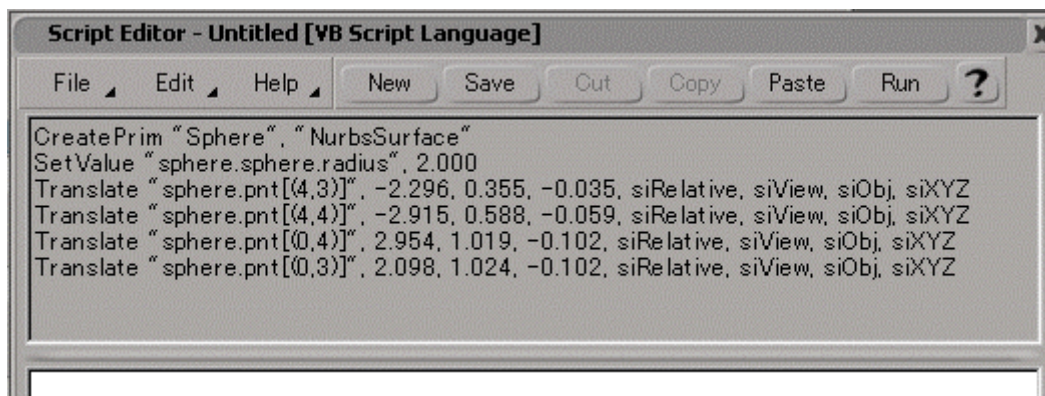
1. スクリプトエディタアイコン (または Alt + 4) をクリックして、Script エディタを開きます。



2. **Edit > Clear History Log** を選択してヒストリをクリアします。
3. **Get > Primitive > Surface > Sphere** を選択し、Sphere を取り出します。
この時、ヒストリには、以下のように表示されます。

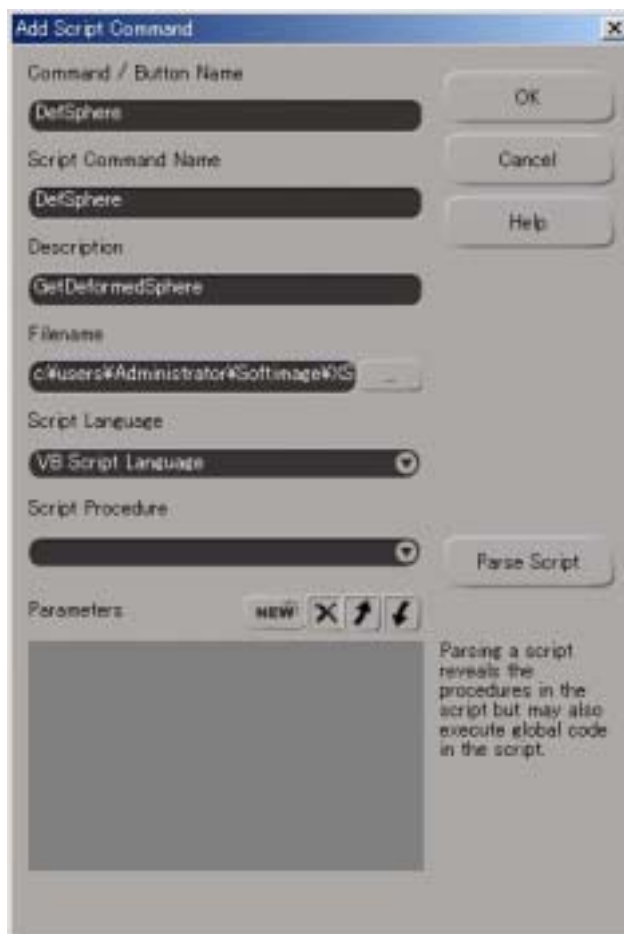
CreatePrim "Sphere", "NurbsSurface"

4. sphere プロパティページにて、**Radius, U Subdivision, V Subdivision** の値を変更してみます。ヒストリには、**SetValue** コマンドとプロパティが表示され、値が設定されているのが分かります。
5. **M** キーを押しながら、任意のビューでポイントを移動させてみます。ヒストリには、**Translate** コマンドが表示されています。ある程度、形状を変えてみて下さい。



6. **View > Custom Toolbars > New Toolbar** を選択し、ツールバーの名前を "MyTools" と入力し、**OK** をクリックします。新しいツールバーが開きました。

7. ヒストリから、CreatePrim から Translate の最後の行までを選択し、ツールバーまでドラッグ&ドロップします。
8. Add Script Command プロパティエディタが開きますので、Command/Button Name に”DefSphere”、Script Command Name にも”DefSphere”と入力します。
注：必要があれば、Filename にもファイル名を設定して下さい。デフォルトでセーブされるディレクトリは、C:\¥users¥<username>¥Softimage¥XSI_<Ver.No>¥Data¥Scripts になります。



9. OK をクリックすれば、MyTools に DefSphere ボタンが登録されます。
10. ここで、登録した DefSphere ボタンをクリックしてみましょう。恐らく、新しい球が表示されると思いますが、変形は行われていません。これは、既にシーン内に Sphere という名前が存在しているため、新しい球の名前が Sphere1 になってしまっているからです。試しに、Edit > Delete All とした後に、DefSphere をクリックしてみてください。今度は、変形まで行われた球が取り出せたはずです。次のステップでは、この Script を編集し、汎用的に使えるようにしましょう。

スクリプトの編集

11. DefSphere ボタンを右クリックし、Edit Script を選択します。Script エディタに内容が

表示されます。

12. **CreatePrim** 行の下に以下の行を追加します。

```
set ob = GetValue( " SelectionList " )(0)
```

説明：*GetValue* は *SOFTIMAGE/XSI* が提供しているコマンドです。引数は取り出したいパラメータですが、ここでは"SelectionList"を指定することにより、選択されているオブジェクト、コンポーネント、パラメータなどを取り出すことができます(*CreatePrim* 直後は自動的に選択状態になっています)。*SelectionList*は、*Collection* (オブジェクトの配列 - *XSI* のスクリプトマニュアル、*Concepts&Techniques for Developers - Selecting through Scripting* を参照)であり、複数のオブジェクトを含むことが可能なので、さらに(0)を指定して、最初のアイテムだけを取り出し、*ob* に代入しています。*GetValue* の仕様や使用例については、*Script Editor* の"?"をクリックして *XSI* の持つコマンド群から参照することができます。

13. **Translate** 行と **SetValue** 行を以下のように変更します。

```
SetValue " sphere.sphere.radius " , XXX
```

```
SetValue ob& ".sphere.radius " , XXX
```

```
Translate " sphere.pnt[(X,X)] " , XXX, XXX, XXX, siRelative, siView, siObj, siXYZ
```

```
Translate ob& ".pnt[(X,X)] " , XXX, XXX, XXX, siRelative, siView, siObj, siXYZ
```

説明：*&*は文字列連結演算子です。*ob* の名前と連結できます。

14. **File > Save** にてセーブし、再度、**DefSphere** をクリックしてみてください。今度は変形した球が取り出せたはずですよ。

ヒント：

- 上記の例では、**SelectionList** を用いて明示的に選択されたアイテムを得ましたが、多くの *XSI* コマンドではデフォルトは選択されているアイテムを対象にしています。従って、対象を明示せずに以下のように省略して記述することも可能です。

```
SetValue " .sphere.radius " , XXX
```

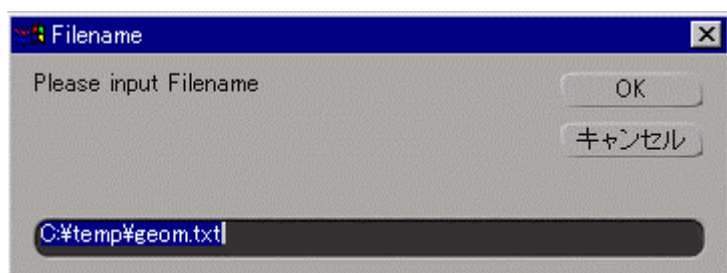
```
Translate " .pnt[(X,X)] " , XXX, XXX, XXX, siRelative, siView, siObj, siXYZ
```
- **Script** エディタ内で、*XSI* コマンドを選択し、**F1** キーを押すとそのコマンドの **Help** が素早く呼び出せます。
- **Script** エディタの **Run** をクリックすることにより、スクリプトを直接実行できます。また、**F5** キーでも **Script** の実行、**CTRL+BREAK** にて中断が行えます。
- 実行したくない部分は ' にてコメントアウトできます。また、行を選択し、**Run** をクリックすればその行のみ実行されます。
- **Script** で実行したことは、通常の操作と同じように **Undo/Redo** ができます。
(**CTRL+Z/CTRL+Y**)
- **File > Keyboard Mapping** にてカスタムコマンドをキーにマッピングすることが可能です。**Group** から **Custom Script Commands** を選択すれば、カスタムコマンドの一覧が表示されますので、コマンドを選択し、キーボード上にドラッグ & ドロップすれば登録されます。

Tutorial 2 : ジオメトリ情報の出力

オブジェクトのジオメトリを書き出してみましよう。

書き出すファイル名を指定する入力ボックスの作成

1. スクリプトエディタアイコンをクリックして、Script Editor を開きます。
2. Edit > Clear History Log を選択して履歴をクリアします。(エディタ領域に文字が残っていれば、Edit > Clear Script Editor も実行して下さい)
3. ファイル名を指定する入力ボックスを作ってみましよう。以下のようにコードを書きます。
`filename = InputBox(" Please input Filename " , " Filename " , " C:¥temp¥geom.txt ")`
 説明 : *InputBox* は VBS 側で提供している文字入力ダイアログボックスです。第 1 引数はメッセージ、第 2 引数はタイトル、第 3 引数はデフォルト値を表しています。その他にも表示位置なども指定できますので詳細は VBS の参考書などで調べてみて下さい。
4. "Run" をクリックしてみます。文字入力ダイアログボックスが表示されました。この段階では、単なる文字列なので、まだファイル名としては認識されていません。"OK" を押してそのまま終了させます。



説明 : この後のチュートリアルで、さらに XSI のダイアログを使ったり、ファイルブラウザなどを用いる例が出てきます。

5. エラートラップを付けて、filename に文字列が入っているときのみ本体のプロシージャ (OutValue) が呼ばれるようにします。以下のようにコードを追加します。

```
If filename <> "" then
    Call OutValue( filename )
End If
```

6. プロシージャ OutValue を記述します。まず、ファイルを扱えるようにするにはファイルシステムのオブジェクトを作成する必要があります。以下のようにコードを追加してみ下さい。

```
Sub OutValue( filename )
    Set FS = CreateObject( " Scripting.FileSystemObject " )
    Set FT = FS.CreateTextFile( filename )
End Sub
```

説明 : *CreateObject* にて、オートメーションオブジェクトを作成します。オートメーションオブジェクトとは、オートメーションインタフェースによって他のアプリケーションやプログラミングツールに公開されるオブジェクトで、*CreateObject* 関数を使用することによって、そのアプリケーションを起動したり、機能呼び出ししたりできます。第 1 引数はアプリケーション名、第 2 引数はクラスです。ここでは、*Scripting*(この場合は *WSH*

の VBS) というアプリケーションの `FileSystemObject` というオブジェクトのクラスを作成しています。その後、`Set` にてオブジェクトへの参照を変数 `FS` に代入します。`CreateTextFile` は、`FileSystemObject` が持つメソッドと呼ばれるもので、ファイル(ファイル名: "filename")を作成し、`TextStream` オブジェクトを返します。`TextStream` オブジェクトは `FT` に代入され、実際に読み書きする際に使用されます。

`CreateObject` 関数、`FileSystemObject`、`CreateTextFile` メソッドなどは Microsoft 社の提供する VBScript ドキュメントや市販の VBS の参考書などで調べることができます。

選択されているモデルのジオメトリ情報の取得とファイル出力

7. 選択されているモデルの情報を取り出してみます。続けて `OutValue` プロシージャ内に以下のコードを追加してみてください。

```
Set mySelList = GetValue( " SelectionList " )
```

説明: Tutorial 1 と同様に `GetValue("SelectionList")` にて選択されているアイテムを取り出します。この段階では、`mySelList` には複数のアイテムが含まれている可能性があることに注意して下さい。

8. 選択されたアイテム数分のループを作り、それぞれの頂点数をファイル出力してみましよう。さらに `OutValue` プロシージャ内に以下のコードを追加します。何かモデルを作成し、選択状態にした後、`Run` してみてください。C:\¥temp¥geom.txt に頂点数が出力されていれば成功です。

```
If mySelList.count > 0 Then
    For i = 0 to mySelList.count - 1
        Set myItem = mySelList.item( i )
        Set myGeometry = myItem.obj
        myNumberOfODGeometries = myGeometry.NbOD
        FT.WriteLine( myNumberOfODGeometries )
    Next
Else
    res = MsgBox( " Nothing Selected. " , 16)
End If
```

説明: `If ~ Then ~ Else ~ End If` は、条件によって処理を分岐させます。`mySelList.count` は `mySelList` に含まれているアイテム数を参照し、`mySelList.item(i)` にて実際のアイテムを取り出します。さらに `myItem.obj` にてジオメトリオブジェクトを取り出し、`myGeometry.NbOD` にて"OD"ジオメトリ数(頂点数)を取り出します(XSI のスクリプトマニュアル、*Programmer's Reference - Properties - Item(SIObjectCollection)*, *Properties - Obj(CollectionItem)*, *Properties - NbOD(Geometry_V1)*を参照)

`WriteLine` は、`TextStream` オブジェクトの持つメソッドで、文字列と改行文字を書き込みます。(VBS マニュアルを参照)

`mySelList.count > 0` でなければ" Nothing Selected." というメッセージボックスが表示されます。`MsgBox` 関数は VBS 側で用意されている関数で、メッセージダイアログボックスを表示させます。第 1 引数はメッセージの内容を、第 2 引数でボタンの種類や個数などを指定できます。(VBS マニュアルを参照)

9. さらに頂点座標を取り出してみましよう。`FT.WriteLine(myNumberOfODGeometries)` の後に以下の行を追加してみてください。

```

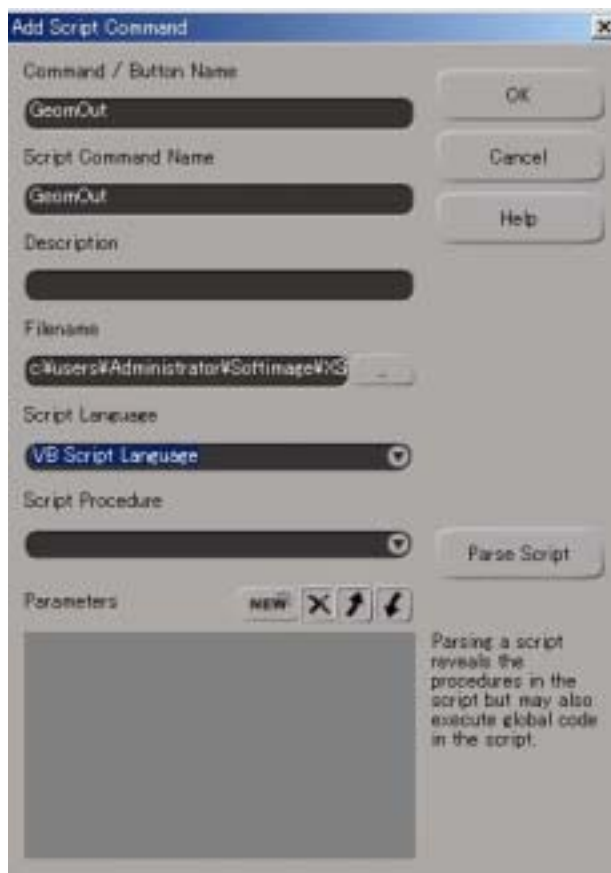
Set myODGeometry = myGeometry.Geometry0D
Set myPos = xsimath.CreateVector3
For j = 0 to myNumberOf0DGeometries    1
    myODGeometry.Position j, myPos
    FT.WriteLine( myPos.x & " " & myPos.y & " " & myPos.z )
Next

```

説明：myGeometry.Geometry0D にて 0D ジオメトリを取り出します。次に頂点座標をストアするオブジェクトを作成します。適切なデータ形式でストアするために、XSI 側の Math Class で提供されているオブジェクト“xsimath.CreateVector3”で、x,y,z の倍精度ベクター型のオブジェクト myPos を確保します。さらに、頂点数分 (myNumberOf0DGeometries) のループで、myODGeometry.Position j, myPos にて、j 番目でインデクスされる、頂点データが myPos 内に得られ、myPos.x, myPos.y, myPos.z にて実際の頂点座標が得られます。（XSI のスクリプトマニュアル、Properties - Geometry0D、Methods - Position(Geometry0D)、Class Hierarchy of the XSI Object Model - Math Classes - SIVector3 - Properties - X/Y/Z を参照）

FT.WriteLine(myPos.x & " " & myPos.y & " " & myPos.z)の&は文字列連結演算子です。

10. Run をクリックし、正常な動作が確認できたら、Edit > Select All にて全てを選択し、MyTools ツールバーにドラッグ&ドロップします。
11. Add Script Command プロパティエディタが開きますので、Command Name に”GeomOut”、Command Name For Scripting にも”GeomOut”と入力します。



注：必要があれば、Filename にもファイル名を設定して下さい。デフォルトでセーブされるディレクトリは、C:¥users¥<username>¥ Softimage¥XSI_<Ver.No>¥Data¥Scripts になります。

- OK をクリックすれば、MyTools に GeomOut ボタンが登録されます。GeomOut ボタンをクリックして最終的な動作確認を行ってみて下さい。

GeomOut.vbs スクリプトの中身

```
filename = InputBox("Please input Filename","Filename","C:¥temp¥geom.txt")
If filename <> "" Then
    Call OutValue( filename )
End If

Sub OutValue( filename )
    Set FS = CreateObject("Scripting.FileSystemObject")
    Set FT = FS.CreateTextFile( filename )
    Set mySelList = GetValue("SelectionList")
    If mySelList.count > 0 Then
        For i = 0 to mySelList.count - 1
            Set myItem = mySelList.item( i )
            Set myGeometry = myItem.obj
            myNumberOfODGeometries = myGeometry.NbOD
            FT.WriteLine( myNumberOfODGeometries )
            Set myODGeometry = myGeometry.GeometryOD
            Set myPos = xsimath.CreateVector3
            For j = 0 to myNumberOfODGeometries - 1
                myODGeometry.Position j, myPos
                FT.WriteLine( myPos.x & " " & myPos.y & " " & myPos.z )
            Next
        Next
    Else
        res = MsgBox("Nothing Selected.", 16)
    End If
End Sub
```

さらに余裕があれば、その他の情報の出力ルーチンやエラートラップなどもインプリメントしてみてください。

カスタムダイアログの作成

- VBS で用意している InputBox ではなく、XSI が用意しているカスタムダイアログに変更してみましょう。以下のように変更してみてください。変更前の行は ' にてコメントにしておくと良いでしょう。

```
filename = InputBox("Please input Filename","Filename","C:¥temp¥geom.txt")
```

```

If filename <> "" Then
    Call OutValue( filename )
End If

```

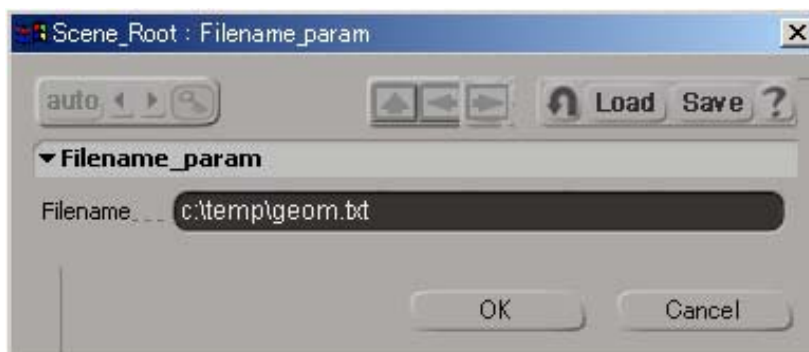
```

SIAddProp "Custom_parameter_list", "Scene_Root", , "Filename_param"
SIAddCustomParameter "Scene_Root.Filename_param", "Filename", siString
SetValue "Scene_Root.Filename_param.Filename", "C:\temp\geom.txt"
On Error Resume Next
InspectObj "Scene_Root.Filename_param", , , siModal
If Err.Number = 0 Then
    filename = GetValue( "Scene_Root.Filename_param.Filename" )
    Call OutValue( filename )
End If
DeleteObj "Scene_Root.Filename_param"

```

説明： *SI_AddProp* コマンドと *SIAddCustomParameter* コマンドは、XSIの持つカスタムパラメータ (*Animate > Create > Parameter > New Custom Parameter Set* と *New Custom Parameter*) で、使用されます。ここでは、*Scene_Root* ノードの真下に *Filename_param* プロパティを持つカスタムパラメータプロパティページを作成しています。*SetValue* コマンドにて、デフォルト値を与え (*SIAddCustomParameter* の引数でデフォルト値を与える方法もあります)、*InspecObj* コマンドにて、プロパティページを開いています。プロパティページにて "OK" がクリックされれば、*Err.Number* が 0 になって返ってくるので (XSI のスクリプトマニュアル、*Commands* の *InspecObj* を参照) *GetValue* コマンドで入力されたファイル名を *filename* に代入し、プロシージャを呼び出します。最後に、*DeleteObj* コマンドにて不要になったカスタムパラメータプロパティページを削除しています。

14. **Run** にて実行してみてください。XSI のカスタムダイアログで入力できるのを確認したら、**File > Save** にてセーブしてください。



ファイルブラウザを使用する

15. VBS にて提供されている **common dialog control (mscomdlg.commondialog)** を用いれば、ファイルブラウザを用いることができます。プロシージャ `Sub OutValue(filename) ~ End Sub` 以外は全てコメントアウトし、以下のコードを追加してみてください。

```

Set DLG = CreateObject("mscomdlg.commondialog")
DLG.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*"
DLG.MaxFileSize = 128

```

```
DLG.FileName = "geom..txt"
DLG.Showopen
If DLG.FileName <> "" Then
    Call OutValue( DLG.FileName )
End If
```

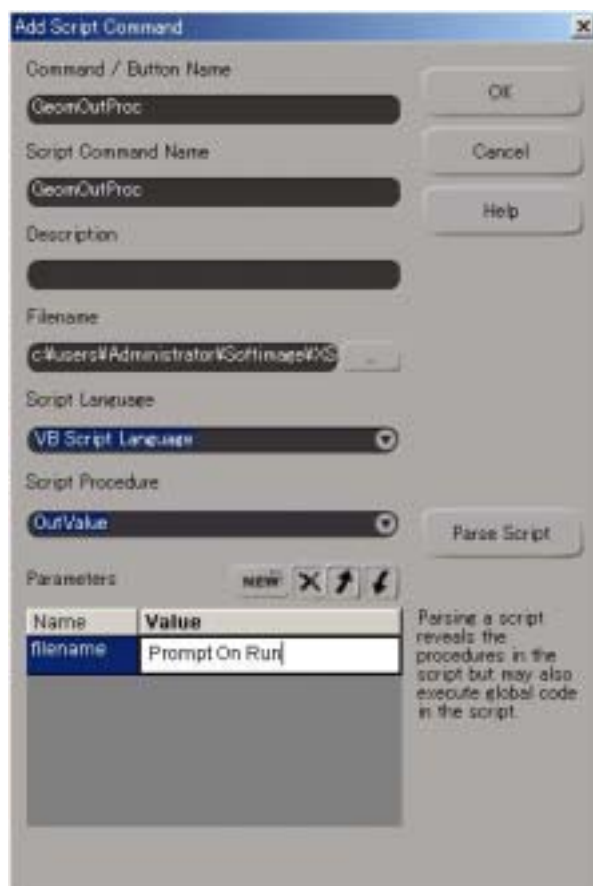
16. Run にて実行してみてください。ファイルブラウザが使用できるのを確認したら、File > Save にてセーブして下さい。



ヒント：DLG.Filter を設定すれば、ファイルの拡張子によるフィルタが使用できます。ブラウジング時に“ファイルの種類”から拡張子を選択すれば、不要なファイルをフィルタリングできます。

グローバルコードを含まない Script の登録

17. プロシージャのみを含む Script を登録し、プロシージャの引数を実行時に与えるようにしましょう。今度は、別のカスタムボタンとして登録してみます。Script エディタからプロシージャ Sub OutValue(filename) ~ End Sub を選択し、ツールバーにドラッグ&ドロップして下さい。Add Script Command ダイアログが開きます。
18. 以下のように入力します。
Command Name : GeomOutProc
Command Name For Scripting : GeomOutProc
Filename : C:\¥users¥<username>\¥Softimage¥XSI_<Ver.No>\¥Data¥Scripts¥GeomOutProc.vbs
19. Parse Script ボタンをクリックします。Script からプロシージャと引数を検出してリストアップしてくれます。Script Procedure には”OutValue”が、Parameters には filename の引数がリストアップされます。
20. Value のボックスをクリック（ボックスの右寄りの部分）して表示されるプルダウンから Prompt On Run を選択します。これによって、この Script の実行時には引数の入力を促す入力ボックスが表示されることになります。



21. OK をクリックすると GeomOutProc ボタンが登録されます。
22. GeomOutProc ボタンをクリックしてみてください。ファイル名を入力するダイアログが表示されます。ファイル名を入力し、OK をクリックするとファイルに出力されます。

