



Technical Note 04-44

COM オブジェクト

By Yvan Ayaay, Technical Support, 4D, Inc.
Technical Note 04-44

(原題: Accessing COM Objects within 4D)

概要

4th Dimension の新しいコマンド LAUNCH EXTERNAL PROCESS を使用すれば COM オブジェクトにアクセスすることが可能になります。今回は、その方法について取り上げたいと思います。実行には CreateObject マクロを使用します。このマクロは、4D Macro Pack に含まれており、COM オブジェクトにアクセスするためのスクリプトを 4D で出力することができます。加えてマクロには COM オブジェクトおよびそのプロパティを表示することのできるブラウザも備わっています。

COM オブジェクトとは？

COM オブジェクトは、データを操作/伝達するためのインスタンス化されたクラスです。COM オブジェクトはファイル、ディスク、ドキュメントといったオブジェクトを表わすソフトウェアオブジェクトで、独自のプロパティやアクセス手段を持っています。COM というのは Common Object Model の頭文字の略で、Microsoft が提唱するコンポーネント型アプリケーションを構築するためのフレームワークです。この規格により、異なるアプリケーションが同じバイナリを共有することが可能になります。VBScript、JScript、C、C++、C# および Visual Basic は、こうしたオブジェクトにアクセスするために使用することができます。COM オブジェクトは、通常、DLL、EXE ファイルのバイナリモジュールである COM ライブラリの中に納められています。たとえば、Excel.Application という COM オブジェクトは、EXCEL.EXE ファイルの COM ライブラリに入っています。Windows システムには、無数の COM オブジェクトがインストールされています。

アクセスする方法

COM オブジェクトは、VBScript などのスクリプトを使用してアクセスすることができます。スクリプトを実行するためにはアプリケーションが必要です。Windows Script Host (WSH) は、スクリプトの実行を制御するための環境を提供します。まとめると、WSH がスクリプトを実行し、スクリプトが COM オブジェクトにアクセスします。最近のバージョンの Windows (XP、2000、2003) には WSH が標準で備わっており、コマンドプロンプトからのスクリプトの実行を可能にしています。インストールされているかを調べるには、コマンドプロンプトを開いて `cscript -?` と入力します。

```
コマンド プロンプト
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Admin04>cscript -?
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

使い方 : CScript scriptname.extension [オプション...] [引数...]

オプション :
//B      バッチ モード : スクリプトのエラーおよびプロンプトを非表示にする
//D      アクティブ デバッグを使用可能にする
//E:engine スクリプト実行時にエンジンを使用する
//H:CScript 既定のスクリプト ホストを CScript.exe に変更する
//H:WScript 既定のスクリプト ホストを WScript.exe に変更する (既定値)
//I      対話モード (既定値、//B と逆の動作)
//Job:xxxx WSF ジョブを実行する
//Logo   ログを表示する (既定値)
//Nologo ログを表示しない : 実行時に見出しを表示しない
//S      このユーザーの現在のコマンド ライン オプションを保存する
//T:nn   秒単位のタイムアウト時間 : スクリプトを実行できる時間の最大値
//X      デバッガでスクリプトを実行する
//U      コンソールからリダイレクトされた I/O に Unicode を使用する

C:\Documents and Settings\Admin04>
```

COM オブジェクトは、インスタンス化することによって、メソッドやプロパティを使用できるようになります。以下の createfolder.vbs という VBScript は、COM オブジェクト Scripting.FileSystemObject をインスタンス化し、そのメソッド CreateFolder を使用してフォルダを作成しています。

```
' File: createfolder.vbs
' Description: This script will use Scripting.FileSystemObject to create a folder.
Dim ArgObj, SArgCount
Set ArgObj = WScript.Arguments
SArgCount=ArgObj.Count
set fso=CreateObject("Scripting.FileSystemObject")    'instantiate Com Object
var1 =fso.CreateFolder(ArgObj(0))                    ' use COM Object method
```

スクリプトをコマンドプロンプトで実行するには、WSH のコマンド CScript か WScript を使用し、スクリプトをアーギュメント、この場合は作成するフォルダのパスとともにコールします。

```
C:\>cscript createfolder.vbs c:\documents\scriptfolder
```

スクリプトが実行されると scriptfolder というフォルダが documents ディレクトリに作成されます。

4D から COM オブジェクトにアクセスする場合、LAUNCH EXTERNAL PROCESS コマ

ンドで外部プロセスとして Windows Script Host (WSH) を起動し、スクリプトを実行してアクセスします。

```
LAUNCH EXTERNAL PROCESS (fileName{; inputStream{; outputStream{;
errorStream}}})
```

fileName パラメータには起動するファイルのパスとアーギュメントを渡します。inputStream と outputStream は任意の入出力ストリームが入ります。errorStream は外部プロセスのエラーストリームです。したがって WSH で VBScript を使用して COM オブジェクトにアクセスしようとする場合、次のようなシンタックスになります。

```
C_TEXT(createfolder_script)
C_TEXT($input;$output)
createfolder_script="cscript.exe %%"
+"c:%%"+"createfolder.vbs%"+"%"c:%%"+"4D%"+"testfolder1%"
LAUNCH EXTERNAL PROCESS(createfolder_script;$input;$output)
```

結果は、コマンドプロンプトからスクリプトを起動した場合と同様のものとなります。

後半では CreateObject マクロの使用を中心に解説をしてゆきます。このマクロは、4D 2004 Macro Pack をインストールすると、メソッドのどこからでも呼び出すことができるものです。使用すると、COM オブジェクトやその内容 (メンバー) を閲覧/選択し、アクセスするための VBScript とプロジェクトメソッドを生成することができます。(このマクロは Windows でのみ使用できます。VBScript を実行するためには、ウイルス監視ソフトのスクリプトブロッキングを解除する必要があるかもしれません。)

CreateObject マクロ

バージョン 2004 では、メソッドエディタのマクロからプロジェクトメソッドを実行できるようになりました。マクロとは、メソッドに挿入することのできるコードの一部分です。マクロはマクロリスト、メソッドメニュー、タイプアヘッドのいずれからでも使用することができます。マクロからプロジェクトメソッドを呼び出すためにタグ<method></method>が使用されます。CreateObject のようなマクロは、この機能を使用して作成されました。

CreateObject を使用するためには、まず 4D 2004 Macro Pack をインストールしなくてはなりません。マクロは 4D Web サイトからダウンロードできます。ファイルに添付されているドキュメントに従って 4D Insider を使用するか、あるいは 4D Insider がいない場合、MacroPack.4DB データベースを起動してメソッド MEM_InstallMacros を実行することによってもマクロをインストールすることができます。

CreateObject マクロを使用する場合の手順は次のようになります。

プロジェクトメソッドを作成します。たとえばメソッド名を FolderCreate とします。メソッドの中で、CreateObject マクロを挿入します。マクロはマクロリスト、メソッドメニュー、タイプアヘッドのいずれからでも使用することができます。

CreateObject マクロが選択されると、オブジェクトブラウザウィンドウが表示されます。COM ライブラリ (通常は DLL か EXE ファイル) を選択すると、そのオブジェクトがリストボックスに表示されます。

リストの表示されたコンポーネントクラスをクリックすると、作成することのできるメンバーが表示されます。たとえば、Scripting ライブラリの FileSystemObject オブジェクトをクリックすると、メンバーとその説明、タイプ、パラメータが表示されます。

OK ボタンをクリックすると、Scripting.FileSystemObject という COM オブジェクトの CreateFolder メソッドにアクセスする VBScript が作成されます。CreateObject マクロを呼び出したメソッドには、その VBScript を LAUNCH EXTERNAL PROCESS でコールするコードが追加されます。

このメソッド FolderCreate は、次のように別メソッドからコールすることができます。

```
`Method CreateNewFolder
`Description: This method will create a new folder on a given folder path
C_TEXT($folderpath)
$folderpath:="C:\¥¥4D¥¥Example"
FolderCreate($folderpath)
```

CreateObject マクロの使用

CreateObject マクロはオブジェクトブラウザ兼スクリプトクリエイターです。ブラウザは、COM オブジェクトのメソッドやプロパティの解析に利用できます。デフォルトのリストのは幾つかの COM ライブラリが挙げられており、また任意の COM ライブラリファイルを選択することも可能です。COM オブジェクトとそのメンバーを選択すると、特定したメソッドあるいはプロパティにアクセスする VBScript とプロジェクトメソッドが生成されます。以下にその使用例をご紹介します。

オブジェクトブラウザ

CreateObject マクロは、オブジェクトブラウザを表示するために TlbInf32.dll ファイルの TypeLib Information Object Library (TLI) を利用しています。マクロがコールされると、プロジェクトメソッドがコールされ、そのメソッドは VBScript を実行して TLI オブジェクトをインスタンス化 (TLI.TypeLibInfo および TLI.Application) します。これらの COM オブジェクトは以下の動作のために使用されます。

デフォルトリストの COM ライブラリファイルを読み込む
ユーザに指定された COM ライブラリファイルを読み込む

COM ライブラリファイルにそのクラスとメンバーを説明した Type ライブラリがあれば、ライブラリのクラス、および各クラスのメンバーをブラウザに表示するために TLI COM オブジェクトが使用されます。

CreateObject マクロがコールするのは MEM_ComOpenObject メソッドです。このメソ

ッドは、MEM_ComStartup メソッドをコールして TLI オブジェクトライブラリをチェックした後、オブジェクトブラウザダイアログを表示します。ユーザが COM オブジェクトとメソッドを選択すると、VBScript にアクセスするためのコードが、マクロを呼び出したメソッドに挿入されます。

CreateObject マクロが使用する VBScript は全部で 6 つあります。

checkTLI.vbs
getLibrary.vbs
getClass.vbs
getobjectname.vbs
getMembers.vbs
getDescription.vbs

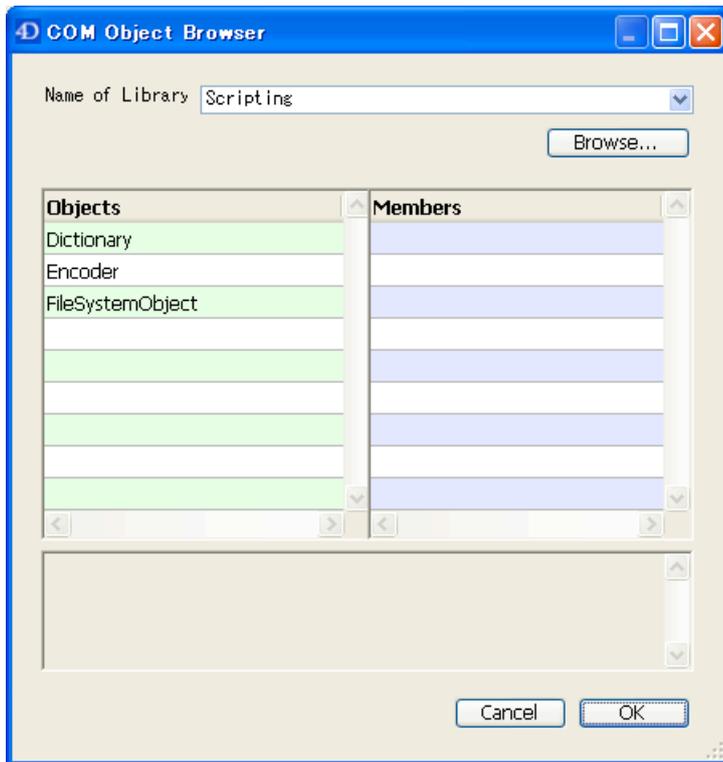
これらのスクリプトは 4D Macro Pack の Scripts フォルダに収められていますので、テキストエディタ等で開いて中身を確認することができます。

checkTLI.vbs は、TypeLib Information Object Library をチェックします。この COM ライブラリがシステムに登録されていない場合は 0、それ以外の場合は 1 を返します。このスクリプトをコールするのは MEM_ComStartup メソッドです。はじめにスクリプトのパスをメソッド MEM_GetScriptPath でチェックします。次にシステムの TLI オブジェクトライブラリを含んでいる TLBINF32.DLL ファイルをコピーします。ここで checkTLI.vbs がコールされ、TLI アプリケーションが登録されているかを調べます。登録されていないなら regsvr32.exe アプリケーションを使用して登録します。



getLibrary.vbs は、ポップアップ/ドロップダウンリストに挙げられる COM ライブラリの名前を取得します。デフォルトリストから選ばれた COM オブジェクトの場合は、ライブラリの COM オブジェクトを使用し、Windows のレジストリを調べて COM ライブラリファイルのパスを取得されます。ブラウズして選ばれたファイルの場合は、すでにパスが分かっています。COM ライブラリファイルのパスに基づき、getClass.vbs は COM ライブラリの作成可能なクラスのリストを取得します。

ブラウザのダイアログ画面がロードされると、getLibrary.vbs、次いで getClass.vbs が実行されます。フォームメソッドでは、はじめに各スクリプトのパスを取得しています。getLibrary.vbs が実行されると、返された値で配列が作られ、ポップアップ/ドロップダウンリスト MEM_lib1 として表示されます。

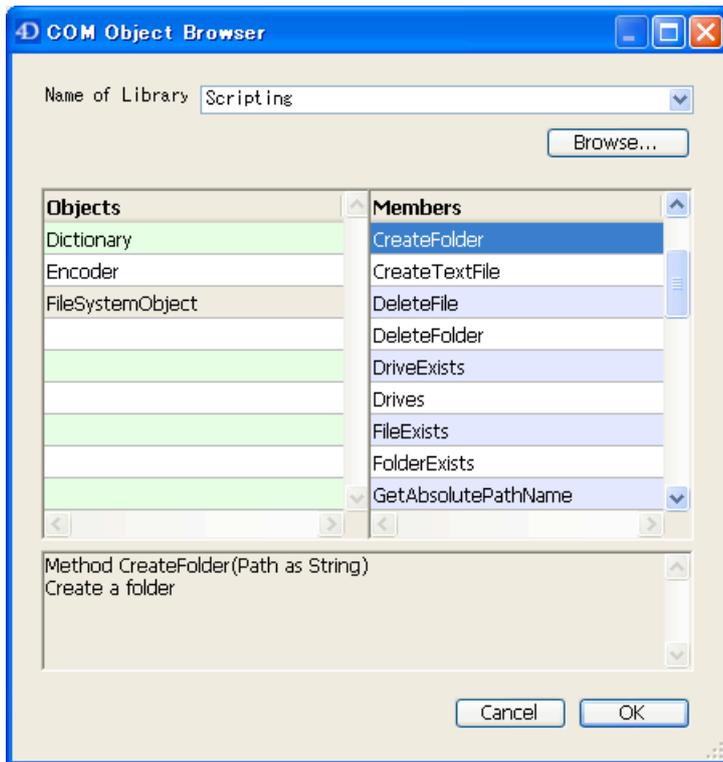


フォームメソッドの続く部分では、MEM_ComGetObjects メソッドをコールしてデフォルトのライブラリオブジェクトやクラスを取得しています。COM ライブラリの作成可能なクラスのリストを取得するために実行されるのが getClass.vbs です。返された値で配列が作られ、リストボックスの列 MEM_obj として表示されます。

getobjectname.vbs は、ブラウズボタンで選択された COM ライブラリファイルのオブジェクト名を取得します。ブラウズボタンのオブジェクトメソッドは、スクリプトを実行し、選択されたライブラリファイルをポップアップ/ドロップダウンリストに追加します。

getMembers.vbs は、選択された COM オブジェクトのメソッドおよびプロパティを取得します。リストボックスの列 MEM_Obj1 のセレクションがクリックされると、列のオブジェクトメソッドによって getMembers.vbs が実行されます。返された値で配列が作られ、リストボックスの列 MEM_prop1 として表示されます。

getDescription.vbs は、COM オブジェクトのメソッドおよびプロパティの説明を取得します。これをコールするのはリストボックスのオブジェクトメソッドです。メソッドあるいはプロパティがクリックされると getDescription.vbs が実行されます。メンバーのタイプ（メソッドあるいはプロパティ）と説明が返され、テキスト変数 MEM_Desc1 として表示されます。



VBScript とメソッド

COM オブジェクトのメソッドあるいはプロパティを選択して OK ボタンをクリックすると、そのオブジェクトにアクセスする VBScript を生成します。OK ボタンのオブジェクトメソッドが MEM_ComObjectMethod をコールすると選択された COM オブジェクトをインスタンス化し、メソッドあるいはプロパティにアクセスする VBScript ファイルが作られます。MEM_ComOpenObject は、CreateObject マクロを呼び出したメソッドに生成された VBScript を実行するコードを追加します。

VBScript を実行するコード

```
C_TEXT($0)
```

```
C_TEXT($1)
```

```
LAUNCH EXTERNAL PROCESS("wscript.exe %%" + MEM_GetScriptPath
+"myscript1.vbs%" + Char(34) + $1 + Char(34); $in; $out) ` CreateFolder($1)
$0:=MEM_ComGetResult
```

VBScript

```
Dim ArgObj, SArgCount
Set ArgObj = WScript.Arguments
SArgCount=ArgObj.Count
set fso=CreateObject("Scripting.FileSystemObject")
  if(SArgCount=1) Then
var1 =fso.CreateFolder(ArgObj(0))
elseif(SArgCount=2) Then
var1 =fso.CreateFolder(ArgObj(0), ArgObj(1))
elseif(SArgCount=3) Then
var1 =fso.CreateFolder(ArgObj(0), ArgObj(1),ArgObj(2))

elseif(SArgCount=4) Then
var1 =fso.CreateFolder(ArgObj(0), ArgObj(1),ArgObj(2),ArgObj(3))
elseif(SArgCount=4) Then
var1 =fso.CreateFolder(ArgObj(0), ArgObj(1),ArgObj(2),ArgObj(3),ArgObj(4))
end if

var2="#"

Set fs=CreateObject("Scripting.FileSystemObject")
Set a=fs.CreateTextFile("C¥4D Macro Pack¥Macro Pack Source¥Scripts¥mytext.txt",
True)

a.WriteLine(var1)
a.WriteLine(var2)
a.Close
```

まとめ

バージョン 2004 にて、4D から COM オブジェクトにアクセスすることができるようになりました。これを可能にしたのは新しいコマンド LAUNCH EXTERNAL PROCESS です。4D Macro Pack に含まれてい CreateObject マクロを使用すれば、COM オブジェクトとそのプロパティやメソッドをブラウズし、それにアクセスするための VBScript や 4D コードを生成することができます。マクロによって起動されるこのブラウザそのものが表示のために COM オブジェクトを使用しています。