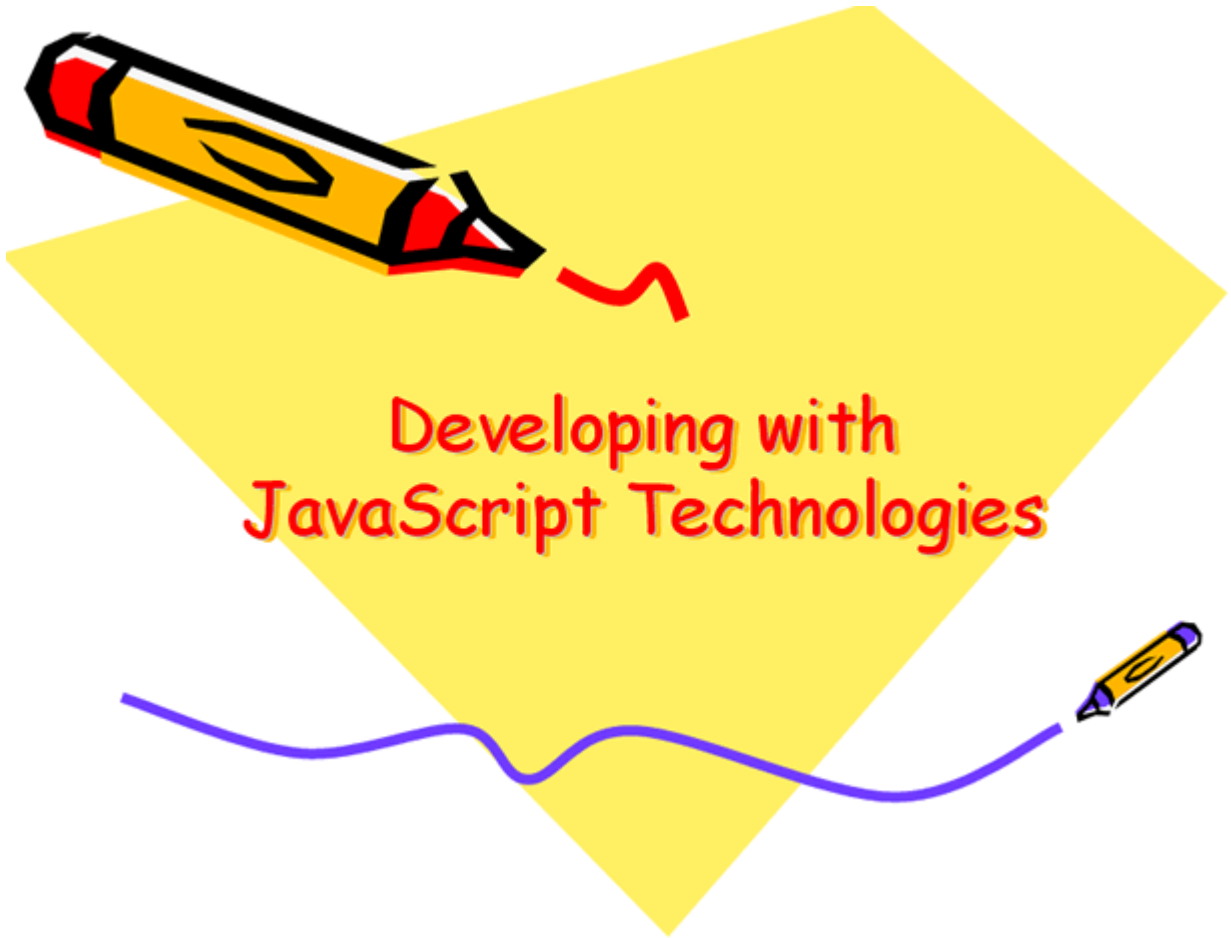
transentia

Exercise 2




# Scripting Windows with JavaScript

# Exercise: Windows Scripting

## Introduction

These exercises will show you how to use JavaScript as a "next generation" shell scripting language under Windows.

These exercises are adapted from examples found at:
http://msdn.microsoft.com/library/en-us/wmisdk/wmi/scripting_in_wmi.asp and
http://www.chegorian.com/jscript/wsh-bas.asp.

## Set Up

Make a new directory for this exercise. Call this directory **shell**:

```
C:\> mkdir shell
C:\> cd shell
```

All the files that you subsequently create as you do this exercise should be contained in this directory.

## Exercise

This exercise will show you how to examine the list of processes running on a system by interfacing with the Windows Management Interface (WMI) facility.

### Create the Script Source

Create the following in a file called **findnotepad.js**:

```
var
  exe = "notepad.exe";
var
  objLocator = new ActiveXObject("WbemScripting.SWbemLocator"),
  objService,
  objEnumerator,
  objInstance,
  i;
var
  e = new Error();

try
  {
  objService = objLocator.ConnectServer(".","root\\cimv2");
  }

catch (e)
  {
  WScript.Quit(1);
  }
```

```
objService.Security_.impersonationlevel = 3;  // normal, non-inheritable

var
  strLogQuery =
    "select Description,Handle from Win32_Process WHERE Caption='" +
      exe +"'";

objEnumerator = new Enumerator(objService.ExecQuery(strLogQuery));

if (objEnumerator.atEnd())
  WScript.Echo("Process is not running on this box");
else
  for (;!objEnumerator.atEnd();objEnumerator.moveNext())
    {
    objInstance = objEnumerator.item();
    with (objInstance)
      WScript.Echo("Process ID: "+Handle+ "\tProcess Name: "+Description);
    }

WScript.Quit(0);
```

***Points of interest***

- The WbemScripting.SWbemLocator ActiveX object, which provides an interface to the WMI system
- The use of the SQL-like WMI Query Language (WQL)

**Execute the Script**

In the Command Prompt window, execute the following commands (the first line opens 4 notepad processes, which are then looked for in the script started as the second command):

```
C:\shell> for %i in ( 1 2 3 4 ) do notepad
C:\shell> cscript /nologo findnotepad.js
```

You should see something similar to the following: