Step-by-Step Tutorials  

#5 - Using WMI

Today's Lesson: "Using WMI" Windows Management Instrumentation

In this lesson we will cover how to use Microsoft's WMI to find out some basic info about our windows computer.

In order to use these Tutorials you will need to purchase DesktopX for $14.95 from Stardock (http://www.stardock.com/products/desktopx/).

Lets get started.

**STEP 1 - Create a text object.**

I'm not going to go back over the first 4 tutorials, if you haven't read them, and gone thru them, please go back and do that first.

Create a TEXT objet, make the default text something like "TEST" it doesn't matter, make it the size you want (big enough to read on your screen).

We are simply going to be using this text object to display the info we gather. Again, this is just to show you HOW to use WMI to retrieve info from your computer, not how to make a pretty widget. We can do that later.

Don't worry about adding scripts yet, we will do that in a later step.

**STEP 2 - WMI Basic connection**

WMI is used to pull information from your system, things like Computer name, User name, What drives are attached to your system. We are going to pull this information from your computer using a simple script in DX.

Open your object, click on NEW for the script.

We will be adding some info to the "Sub Object_OnScriptEnter"

```vbscript
Dim objWMIService
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
Set Info = objWMIService.ExecQuery ("Select * from Win32_ComputerSystem")
```

The DIM is used to Declare variables and allocates storage space.

The first SET command is used to make the connection to the local computer. I could get all technical on this part, but if you really want to know all the details on this connect string visit this site.

The 2nd SET command is used to pull the information you want from the computer. In this example we are wanting to get the user name, domain name, and computer name. This works like a SQL Query string, there are a ton of options for this as well. Again, this is a BASIC tutorial, so if you want more details, look the commands up.

Let's break this line apart:

- Set CompInfo
- Set the variable CompInfo to the info we pull from WMI
- objWMIService
- objWMIService was setup in the previous SET command to point to the WMI service.
- .ExecQuery ( 
- Execute a query into the WMI database
"Select *" we want to "SELECT" * - EVERYTHING
we could have picked just 1 item if we only wanted say the Username, etc.
from Win32_ComputerSystem") The data we want is coming from the Win32_ComputerSystem part of WMI
There are DOZENS of these tables in WMI.

HERE is a great place to learn all about WMI

Lets look at what information we can pull from this one table:
I'm not going to go through ALL of this, I just want you to see what is available from this 1 table.

class Win32_ComputerSystem : CIM_UnitaryComputerSystem
{
    uint16 AdminPasswordStatus;
    boolean AutomaticManagedPagefile;
    boolean AutomaticResetBootOption;
    boolean AutomaticResetCapability;
    uint16 BootOptionOnLimit;
    uint16 BootOptionOnWatchDog;
    boolean BootROMSupported;
    string BootupState;
    string Caption;
    uint16 ChassisBootupState;
    string CreationClassName;
    sint16 CurrentTimeZone;
    boolean DaylightInEffect;
    string Description;
    string DNSHostName;
    string Domain;
    uint16 DomainRole;
    boolean EnableDaylightSavingsTime;
    uint16 FrontPanelResetStatus;
    boolean InfraredSupported;
    string InitialLoadInfo;
    datetime InstallDate;
    uint16 KeyboardPasswordStatus;
    string LastLoadInfo;
    string Manufacturer;
    string Model;
    string Name;
    string NameFormat;
    boolean NetworkServerModeEnabled;
    uint32 NumberOfLogicalProcessors;
    uint32 NumberOfProcessors;
    uint8 OEMLogoBitmap[];
    string OEMStringArray[];
    boolean PartOfDomain;
    sint64 PauseAfterReset;
    uint16 PCSystemType;
    uint16 PowerManagementCapabilities[];
    boolean PowerManagementSupported;
    uint16 PowerOnPasswordStatus;
    uint16 PowerState;
    uint16 PowerSupplyState;
    string PrimaryOwnerContact;
    string PrimaryOwnerName;
    uint16 ResetCapability;
    sint16 ResetCount;
    sint16 ResetLimit;
    string Roles[];
    string Status;
    string SupportContactDescription[];
    uint16 SystemStartupDelay;
    string SystemStartupOptions[];
    uint8 SystemStartupSetting;
    string SystemType;
    uint16 ThermalState;
    uint64 TotalPhysicalMemory;
string UserName;
uint16 WakeUpType;
string Workgroup;
;
What we want from this table is:

- **UserName** - User Name
- **TotalPhysicalMemory** - Total Memory
- **Domain** - Domain name (If there is one)
- **Name** - Computer Name

**STEP 3 - Pulling info from WMI**

So, we know how to make the connection to WMI, and we know what table we want. Lets get the USERNAME that is logged into the current computer.

For Each objComputer In CompInfo
    PCName = objComputer.Name
Next
object.text = "ComputerName: " & PCName

Let's look over this one too.

I'm not going to break everything apart, but the basics are this:
the FOR EACH is used to pull 1 item at a time from the COMPINFO we set before, and store it in the objComputer variable.
Then we assigning the variable PCName to the .NAME field from the table.
Then we take and set the TEXT of the object to "ComputerName: " & PCName

SAVE & APPLY this now, and see what happens.
What you SHOULD see is your text object showing something like:
ComputerName: MyPcsName

Not to hard was it? Let's add a little more.

**STEP 4 - Adding more info**

Open the object back up, EDIT the script. Let's add a little more info.

For Each objComputer In CompInfo
    PCName = objComputer.Name
    PCDomain = objComputer.Domain
    UserName = objComputer.UserName
Next
object.text = "ComputerName: " & PCName & vbCrLf
object.text = object.text & "Domain: " & PCDomain & vbCrLf
object.text = object.text & "UserName: " & UserName & vbCrLf

We added the Domain name (if there is one, you may not have one), and the Username.
Notice the USERNAME has the domain name as part of it? (if you have a domain, otherwise you wont see this).
We want to pull JUST the user name out from that. So lets add a simple IF statement to the script.

If len(PCName) > 1 Then
    t = len(PCName)+2
    UserName = mid(UserName,t)
End If
Insert this script right below the **UserName = objComputer.UserName** line. This simply looks at the **PCName** to see if its **LEN**gth is over 1 character. Then it gets the length of that name + 2 spaces (the \ and then the first letter of the name). Then it sets **UserName** to the MIDdle of the the string **UserName** starting at the T position.

Don't worry I will put the entire script at the bottom in one place, so you can copy/paste it if you want.

**STEP 5 - What Drives do we have?**

We are going to add a little more to this script. One of the things people have asked to see is how to determine what drives, drive letters, etc. are on your pc.

The code we are going to add looks a lot like what we used above:

```vba
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
Set DriveInfo = objWMIService.ExecQuery("Select * from Win32_LogicalDisk")
object.Text = object.Text & vbCrLf & "DeviceID | DriveType | Description " & vbCrLf & object.Text & " | VolumeName | FreeSpace | FileSystem"
For Each objDrive In DriveInfo
    object.Text = object.Text & vbCrLf & objDrive.DeviceID & " | " & vbCrLf & objDrive.DriveType & " | " & vbCrLf & objDrive.Description & vbCrLf & & vbCrLf & object.Text & vbCrLf & objDrive.VolumeName & " | " & vbCrLf & objDrive.FreeSpace & " | " & vbCrLf & objDrive.FileSystem
Next
```

We are pulling a lot of info from the **WIN32_LogicalDisk** table. add the code, see what shows up. You should see something like:

<table>
<thead>
<tr>
<th>DeviceID</th>
<th>DriveType</th>
<th>Description</th>
<th>VolumeName</th>
<th>FreeSpace</th>
<th>FileSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:</td>
<td>3</td>
<td>Local Fixed Disk</td>
<td>Boot-2006b</td>
<td>30640107520</td>
<td>NTFS</td>
</tr>
<tr>
<td>D:</td>
<td>3</td>
<td>Local Fixed Disk</td>
<td>Storage-2006b</td>
<td>14589018112</td>
<td>NTFS</td>
</tr>
<tr>
<td>E:</td>
<td>5</td>
<td>CD-ROM Disc</td>
<td>20030803_1304</td>
<td>0</td>
<td>CDFS</td>
</tr>
<tr>
<td>F:</td>
<td>5</td>
<td>CD-ROM Disc</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**STEP 5 - Here's The code:**

Promised to post all the code in one place, here it is:

```vba
'Called when the script is executed
Sub Object_OnScriptEnter
    Dim objWMIService
    Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
    Set CompInfo = objWMIService.ExecQuery("Select * from Win32_ComputerSystem")
    For Each objComputer In CompInfo
        PCName = objComputer.Name
        PCDomain = objComputer.Domain
        UserName = objComputer.UserName
        If Len(PCName) > 1 Then
            t = Len(PCName)+2
            UserName = Mid(UserName, t)
        End If
        object.text = "ComputerName: " & PCName & vbCrLf & object.text & "Domain: " & PCDomain & vbCrLf & object.text & "UserName: " & UserName & vbCrLf &
        Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
        Set DriveInfo = objWMIService.ExecQuery("Select * from Win32_LogicalDisk")
        object.Text = object.Text & vbCrLf & "DeviceID | DriveType | Description " & vbCrLf & object.Text & " | VolumeName | FreeSpace | FileSystem"
        For Each objDrive In DriveInfo
            object.Text = object.Text & vbCrLf & objDrive.DeviceID & " | " & vbCrLf & objDrive.DriveType & " | " & vbCrLf & objDrive.Description & vbCrLf & & vbCrLf & object.Text & vbCrLf & objDrive.VolumeName & " | " & vbCrLf & objDrive.FreeSpace & " | " & vbCrLf & objDrive.FileSystem
        Next
    Next
End Sub
```
CONCLUSION

There are a lot of things that can be done with this info. Things like making a Drive monitor, or a simple piece of text with your username etc on it.
WMI is EXTREMELY powerful, you can pull things from other computers on your network, A simple GOOGLE search will show you TONS of places with lots of script examples.

I hope you have enjoyed this step into DX, and look forward to the next installment.