

WMI ?

- WMI とは Windows Management Instrumentation の略
 - Windows Driver Model の拡張の一種
- Windows が管理する情報へのインターフェース
 - 例えば CPU の情報であったり、物理ハードディスクの情報などが取れます
- WMI の各クラスにはプロパティ・メソッド・イベントがある
 - 今回はプロパティの取得方法を紹介します。

WMI Classes

- 代表的な Class は
 - Win32_Process
 - Win32_OperatingSystem
 - Win32_ComputerSystem
 - Win32_Processor
 - Win32_Product
 - Win32_BIOS

などなど.....

使い方

- JScript や VBScript では SQL に似た構文で取得可能です
 - この構文を WQL (WMI Query Language) といいます
 - WQL を使わない取得方法もあります(次ページ)

【JScript での例】

```
var locator = WScript.CreateObject("WbemScripting.SWbemLocator");
var service = locator.ConnectServer();
var processorInfo = service.ExecQuery("SELECT * FROM Win32_Processor");

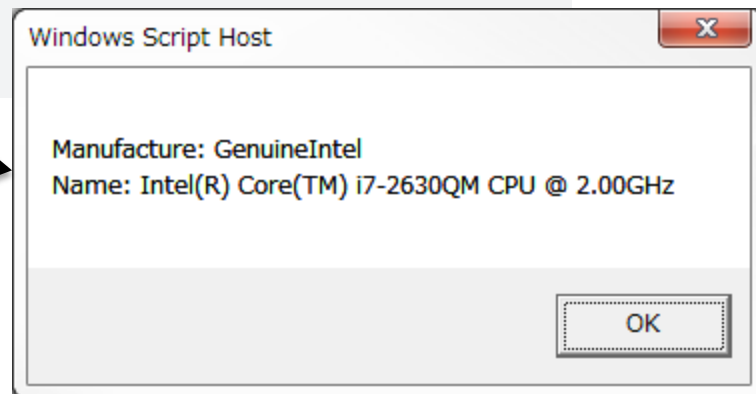
var e = new Enumerator(processorInfo);

var info = "";

while (!e.atEnd()) {
    var item = e.item();
    info += "Manufacture: " + item.Manufacturer + "¥nName: " + item.Name + "¥n¥n";
    e.moveNext();
}

WScript.Echo(info);
```

Echo で表示された内容



使い方

Delphi でも WQL を使えますが、ここでは普通にプログラムで取得する方法を紹介します。


1. IWbemServices インターフェースを実装したオブジェクトを取得する

- Wbem とは Web-Based Enterprise Management の略
 - 分散コンピューティング環境の管理を統合するために開発されたシステム管理技術群の名称 (wikipedia より)

2. 取得した Services に欲しいクラスの名前を渡して EnumObject を取得する

3. EnumObject から値を取得する

WMI Property の取得

```
function GetWMIProperty(  
    const iClassName: String;  
    const iPropName: array of String;  
    const iOnGetProp: TInnerGetPropEvent;  
    const iAppData): Boolean;  
var  
    WbemServices: IWbemServices;  
    EnumWbemClassObject: IEnumWbemClassObject;  
begin  
    Result := False;  
  
    // Get WbemServices   
    if (not GetWbemServices(WbemServices)) then  
        Exit;  
  
    // Get EnumWbem  
    if  
    (  
        Failed(  
            WbemServices.CreateInstanceEnum(  
                PWideChar(WideString(iClassName)),  
                WBEM_FLAG_SHALLOW or WBEM_FLAG_FORWARD_ONLY,  
                nil,  
                EnumWbemClassObject)  
        )  
    )  
    then  
        Exit;  
  
    Result := EnumObjects(EnumWbemClassObject, iPropName, iOnGetProp, iAppData);  
end;
```

WbemServices の取得

```
function GetWbemServices(  
    var oWbemServices: IWbemServices): Boolean;  
var  
    WbemLocator: IWbemLocator;  
begin  
    Result := False;  
  
    // Get WbemLocator の取得  
    if  
        (  
            Failed(  
                CoCreateInstance(  
                    CLSID_WbemLocator,  
                    nil,  
                    CLSCTX_INPROC_SERVER,  
                    IID_IWbemLocator,  
                    WbemLocator)  
            )  
        )  
    then  
        Exit;
```

```
// Get WbemServices の取得  
if  
    (  
        Failed(  
            WbemLocator.ConnectServer(  
                'root¥cimv2',  
                nil,  
                nil,  
                nil,  
                0,  
                nil,  
                nil,  
                oWbemServices)  
        )  
    )  
then  
    Exit;  
  
    Result := True;  
end;
```

EnumObjects の取得

```
function EnumObjects(  
    const iEnumWbemClassObject: IEnumWbemClassObject;  
    const iPropNames: array of String;  
    const iOnGetProp: TInnerGetPropEvent;  
    const iAppData): Boolean;  
var  
    WbemObjects: array [0.. WBEM_MAX_OBJECT_NESTING - 1] of IWbemClassObject;  
    COMResult: HRESULT;  
    PropHolder: OleVariant;  
    Prop: POleVariant;  
    Count: Cardinal;  
    i, j: Integer;  
    Name: String;  
begin  
    Result := False;  
  
    Prop := @PropHolder;
```


WMI Property の取得

```
while
(
    Succeeded(
        iEnumWbemClassObject.Next(
            WBEM_INFINITE,
            WBEM_MAX_OBJECT_NESTING,
            WbemObjects[0],
            Count
        )
    )
)
do begin
    if (Count = 0) then
        Break
    else
        Result := True;

    for i := 0 to Count - 1 do
        for j := 0 to High(iPropNames) do begin
            Name := iPropNames[j];

            WbemObjects[i].Get(PWideChar(WideString(Name)), 0, Prop, 0, 0);
            iOnGetProp(Name, Prop^, iAppData);
        end;

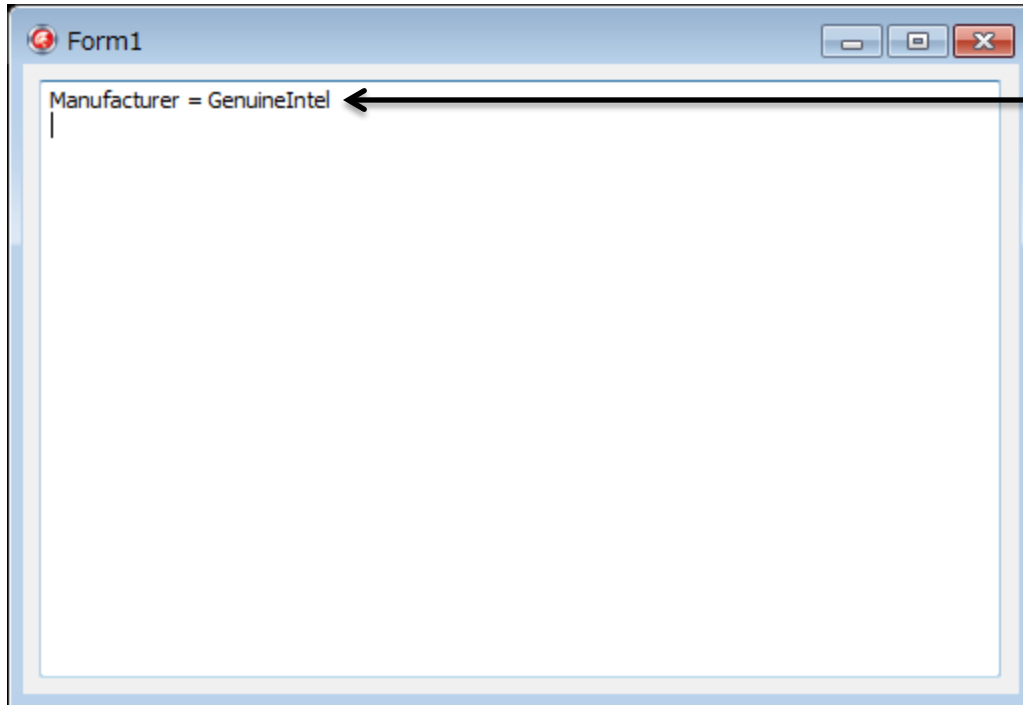
    COMResult := iEnumWbemClassObject.Skip(WBEM_INFINITE, Count);

    if (Failed(COMResult)) or (COMResult = S_FALSE) then
        Break;
end;
end;
```

Processor の製造者名を取ってみる

```
procedure GetPropEvent(const iName: String; const iPropVal: Variant);  
begin  
  Form1.Memo1.Lines.Add(iName + ' = ' + iPropVal);  
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  GetWMIProperty('Win32_Processor', ['Manufacturer'], GetPropEvent);  
end;
```



何に使うのか？

PC 環境の正確な把握

顧客のPCや社内PCの状態の把握

社内システムの管理者など

【Get WMI Information の設定ファイル】

Win32_Processor.Name

Win32_Processor.Manufacturer

Win32_Processor.MaxClockSpeed

Win32_LogicalDisk.Caption

Win32_LogicalDisk.VolumeSerialNumber

Win32_BIOS.Version

Win32_BIOS.SerialNumber

Win32_BaseBoard.Version

Win32_BaseBoard.SerialNumber

Win32_SystemEnclosure.Version

この設定ファイルに記載されているプロパティ
を表示する



The screenshot shows a window titled "Get WMI Information" with a close button in the top right corner. The window contains a list of system properties and their values:

- Win32_Processor.Name: Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz
- Win32_Processor.Manufacturer: GenuineIntel
- Win32_Processor.MaxClockSpeed: 2001
- Win32_LogicalDisk.Caption: C:
- Win32_LogicalDisk.Caption: D:
- Win32_LogicalDisk.VolumeSerialNumber: [Redacted]
- Win32_LogicalDisk.VolumeSerialNumber: [Redacted]
- Win32_BIOS.Version: DELL - 2
- Win32_BIOS.SerialNumber: [Redacted]
- Win32_BaseBoard.Version: A00
- Win32_BaseBoard.SerialNumber: [Redacted]
- Win32_SystemEnclosure.Version: 0.1

At the bottom of the window, there are two buttons: "Report" and "Close". A red arrow points to the "Report" button.

WMI の結果を HTML 化

WMI REPORT 2012/05/30 18:15:03

Win32_Processor.Name	Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz
Win32_Processor.Manufacturer	GenuineIntel
Win32_Processor.MaxClockSpeed	2001
Win32_LogicalDisk.Caption	C:
Win32_LogicalDisk.Caption	D:
Win32_LogicalDisk.VolumeSerialNumber	[REDACTED]
Win32_LogicalDisk.VolumeSerialNumber	[REDACTED]
Win32_BIOS.Version	DELL - 2
Win32_BIOS.SerialNumber	[REDACTED]
Win32_BaseBoard.Version	A00
Win32_BaseBoard.SerialNumber	[REDACTED]
Win32_SystemEnclosure.Version	0.1

by GetWMIInfo Copyright (C) [SERIALGAMES Inc.](#) 2004.

参考文献

- **WMI Fun**
 - WMI の詳しい解説があるウェブページ
 - <http://www.wmifun.net/>
- **WMI Delphi Code Creator**
 - 自動的に Delphi/FreePascalCompiler/prism/C++Builder 用の WMI コードを生成できるツール(情報提供: 福士様)
 - <http://theroadtodelphi.wordpress.com/wmi-delphi-code-creator/>

```
13 MsmcComputer := LocalHost;
14 MsmcFlagForwardOnly := $00000000;
15 var
16   FWMILocator : OLEVariant;
17   FWMIService : OLEVariant;
18   FWMIBaseObject : OLEVariant;
19   FWMIObj : OLEVariant;
20   ofnum : Integer;
21   ofname : String;
22   ofvalue : LongWord;
23 begin
24   FWMILocator := CreateOLEObject('MsmcScripting.SMLocator');
25   FWMIService := FWMILocator.ConnectServer(MsmcComputer, 'root\CIMV2', MsmcUser,
26   FWMIBaseObject := FWMIService.ExecQuery('SELECT * FROM Win32_Process', MS_Where);
27   ofnum := 0;
28   while ofnum <= (FWMIBaseObject.Count - 1) do
29   begin
30     FWMIObj := FWMIBaseObject.Item(ofnum);
31     ofname := FWMIObj.Name;
32     ofvalue := FWMIObj.Value;
33     WriteLn(Format('Command Line %s', [String(FWMIObj.CommandLine)]));
34     WriteLn(Format('Creation Class Name %s', [String(FWMIObj.CreationClassName)]));
35     WriteLn(Format('Description %s', [String(FWMIObj.Description)]));
36     WriteLn(Format('Executable Path %s', [String(FWMIObj.ExecutablePath)]));
37     WriteLn('');
38     ofnum := ofnum + 1;
39   end;
40   FWMIObj := Unassigned;
41 end;
```

何コレ!
カッコ良い!