

## WMI で何が出来るのか

WMI は Windows 上でいろんな情報を取得することが出来る機能です。  
どんな情報が取れるかというと、

- ・ CPU 名、CPU 数、L2 キャッシュサイズ、クロック周波数、など・・・
- ・ 物理メモリ量、仮想メモリ量、物理メモリ使用量、メモリー枚数など・・・
- ・ NIC 名、NIC 枚数、NIC のマックアドレス、IP アドレス、サブネットマスク、リンク速度、など・・・
- ・ ネットワーク利用率、送信バイト数、受信バイト数、など・・・
- ・ OS のバージョン、コンピュータ名、ドメイン名、言語、など・・・
- ・ ターミナルサービスの、セッション数、ログインユーザーのメモリ使用量、ログイン時間、など・・・

自分がいじった事があるのは、このへんですが、他にも、セキュリティソフトの情報だとか、インストールされてるソフトの情報だとか、・・・なんかもう、いろいろ取得出来ます。

もちろん、API を叩けば取得できなくも無いんですが、WMI 一つで、こんな詳細な情報を取得できるわけです。

特に C# の場合は、API 叩くのが面倒なので、その点、WMI は素敵です。

実際に、どんなデータが取得できるのかを見たい場合は、WMI で情報取得するツールのコードをコンパイルして、実行してみてください。

あと、この辺とか参考になるかも。

[http://www.anchor-systems.co.jp/anchor/ashp/netmon/samples/wmi\\_os.html](http://www.anchor-systems.co.jp/anchor/ashp/netmon/samples/wmi_os.html)

## WMI のメモ

WMI の使い方の流れとしては、

1. クラス（情報のかたまり？）を取得
2. クラスから、コレクション（データの配列みたいな）を取得
3. コレクションから一つオブジェクトをとりだす。
4. オブジェクトのプロパティを取得する。

こんな感じ。

コレクションに複数データがあるのは、  
例えば CPU が複数ある場合・・・。

1. CPU の情報を取得するクラスを生成
2. CPU 一覧を取得
3. CPU 一覧から、1つ目を取得
4. 取得した1つ目の CPU 情報の詳細を見る。

というイメージで、一覧を取得したら、繰り返し処理する可能性がある。  
CPU が一つしか無いマシンであれば、CPU 一覧にデータは一つしか無い。

## 実際にコード

Win32\_Processor クラスを使用して、CPU 一覧の情報をコンソールに出力するサンプル (抜粋)

参照の追加で、「System.Management」を追加して、

```
using System.Management;

// クラスを取得
ManagementClass mc = new ManagementClass("Win32_Processor");
// CPU 一覧を取得
ManagementObjectCollection moc = mc.GetInstances();

Console.WriteLine("-----");
// CPU 情報を一つずつ取り出す
foreach (ManagementObject mo in moc)
{
    // 情報を表示
    Console.WriteLine("DeviceID      = {0}", mo["DeviceID"]);
    Console.WriteLine("Name        = {0}", mo["Name"]);
    Console.WriteLine("MaxClockSpeed = {0}", mo["MaxClockSpeed"]);
    Console.WriteLine("L2CacheSize  = {0}", mo["L2CacheSize"]);
    Console.WriteLine("-----");
}
```

こうすると、CPU の情報一覧が表示されると思う。

ただ、Win32\_Processor での情報取得処理は、なぜか時間がかかる。  
理由は良く知らんが、情報は取れてるから問題ないだろう。

他にどんな情報が取れるかは、[WMIで情報取得するツール](#)を参照。

## その他メモ

### ローカルのドライブを列挙する

#### 注意

.NET Framework 2.0 から、[System.IO.DriveInfo](#) というクラスが出来ました。

こいつを使うと、WMI を触らずに、しかも簡単にハードディスクの空き容量などを取得することが出来ます。

なので、ドライブの情報を取るために、無理に WMI を使う必要はありません。

#### 使用するクラス

Win32\_LogicalDisk

#### ドライブの種類

MediaType ? DriveType ?

- DriveType
  - 2 3.5 インチフロッピードライブ
  - 3 ハードディスクドライブ
  - 5 CD-ROM ドライブ
- MediaType
  - 5 3.5 インチフロッピードライブ
  - 12 ハードディスクドライブ
  - 11 CD-ROM ドライブ

ドライブ番号

DeviceID

ファイルシステム

FileSystem

全容量

Size

ディスクが無かったら null

空き容量

FreeSpace

ディスクが無かったら null

その他

これの注意点としては、フロッピーの容量を調べようとして(？)、その結果フロッピーにアクセスしようとしています。CDも同様。

ManagementObjectSearcher を使って、

```
SELECT * FROM Win32_LogicalDisk WHERE DriveType <> 2
```

ってしても、一緒。

どーしてもフロッピーにアクセスしてしまうみたい。

ローカルのハードディスクだけを列挙するだけなら、Win32\_DiskDrive を使うっぼ？