第3章 障害管理ツールの開発

本章では，ネットワーク上のコンピュータに障害がおきた場合などの切り分け作業に有用な，障害管理ツールの開発方法について説明します．

3.1 レジストリ収集

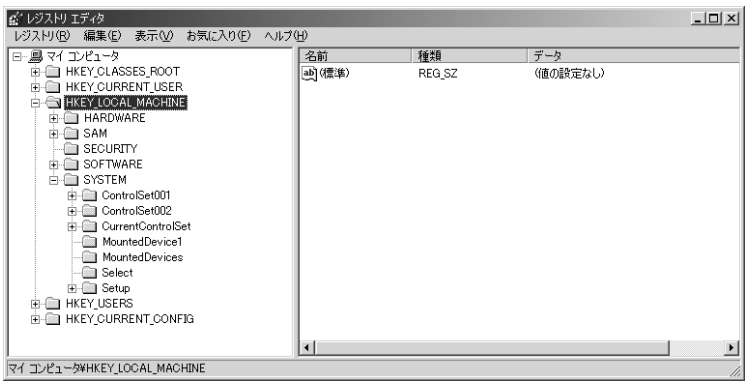
ハードウェア，ソフトウェア資産管理の前に，ネットワーク(LAN)上のコンピュータを管理する方法について説明します．

3.1.1 レジストリの概要

●概要

レジストリは，Windows(OS)の構成情報などを格納するためのローカルデータベースです．通常のユーザーは意識しませんが，システム管理者・ネットワーク管理者はレジストリ自体や，レジストリに何が設定されているかは意識する必要があります．レジストリの参照・変更は，レジストリエディタ(regedit.exeまたはregedt32.exe)を使用します．どちらのレジストリエディタも，Windowsのプログラムメニューには存在せず，「ファイル名を指定して実行」ダイアログから起動します．regedit.exeの表示例を，**図3-1**に示します．

図3-1　レジストリエディタの例



レジストリエディタを使用すれば，ローカルコンピュータだけでなくネットワーク上の他コンピュータのレジストリを参照・変更することもできます．regedit.exeの場合は，［レジストリ］→［ネットワークレジストリへの接続］で，他コンピュータのレジストリに接続できます．

レジストリエディタのregedit.exeとregedt32.exeは，ほぼ同等な操作ができますが，Windows 2000以前のWindowsでは，regedt32.exeでしか扱えないデータ型(REG_MULTI_SZ)があるなど，若干の機能差がありますので，用途に応じて使い分けてください．Windows XPでは，regedit.exeでもREG_MULTI_SZ型を扱えるように改善されています．

●レジストリの構造

レジストリは，4つのレベルで構成されています．レジストリを構成するレベルを，**表3-1**に示します．レベルは上位から，ルートキー→キー→サブキー→値の順で，レジストリエディタ等で表示するとわかりますが，エクスプローラでフォルダ・ファイルを表示した時のように，階層的に整理されています．

表3-1　レジストリの構造

レベルの名称	説　明
ルートキー	システムであらかじめ定義されたレジストリの分類 HKEY_で始まる名称で，5つのルートキーがある
キー	システム定義，ユーザー定義の2種類がある ルートキーのサブディレクトリとして存在 キーとサブキーには関連付けられたデータはなく，階層を整理し，レジストリアクセスを容易にする目的で存在
サブキー	システム定義，ユーザー定義の2種類がある キーのサブディレクトリとして存在
値	レジストリの階層構造の末端に位置し，ファイルシステムのファイルに相当する コンピュータやアプリケーションの設定などに関する実際のデータが設定されている 値には，9種類のデータ型がある

●レジストリ・データ型

レジストリの値には，9つデータ型があります．データ型の名称と説明を，**表3-2**に示します．文字列データの場合は「REG_SZ」，数値の場合は「REG_DWORD」が，よく使用されているデータ型です．

●レジストリ・ルートキー

レジストリには，5つルートキーがあります．ルートキーの名称と説明を，**表3-3**に示します．ルートキーは，OSやハードウェア情報を保存する「HKEY_LOCAL_MACHINE」と，ユーザー情報を保存する「HKEY_USERS」の2つがメインのルートキーで，それ以外のルートキーはメインのルートキー配下にあるサブキーの別名です．

表3-2　レジストリ・データ型

データ型の名称	形　式	説　明
REG_SZ	文字列	テキスト文字列
REG_EXPAND_SZ	文字列	変数を含むテキスト文字列
REG_BINARY	バイナリ	バイナリデータ
REG_DWORD	数値	4バイト長の数値データ
REG_DWORD_LITTLE_ENDIAN	数値	4バイト長の数値データ Intelベースのコンピュータにおけるデータの格納順をリトルエンディアンと呼ぶ．リトルエンディアンの場合，右端が最上位バイト
REG_DWORD_BIG_ENDIAN	数値	4バイト長の数値データ Macintosh，UNIXなどIntel以外のコンピュータにおけるデータの格納順をビッグエンディアンと呼ぶ．ビッグエンディアンの場合，左端が最上位バイト
REG_LINK	文字列	システムとレジストリの値の間のシンボリックリンク
REG_MULTI_SZ	複数行文字列	文字列の配列．複数行の文字列が定義可能
REG_FULL_RESOURCE_DESCRIPTOR	文字列	ハードウェアコンポーネントのリストを格納するための配列

表3-3　レジストリ・ルートキー

ルートキーの名称	説　明
HKEY_CLASSES_ROOT (HKCR)	コンピュータ上に登録されているCOMサーバーの情報，アプリケーションとファイル(拡張子)の関連付け情報を保存するレジストリ HKLM¥Software¥Classesキーの別名
HKEY_CURRENT_USER (HKCU)	コンピュータに現在ログオンしているユーザーの情報(プロファイル)を保存するレジストリ 環境変数，デスクトップの設定，ネットワーク接続，プリンタ，アプリケーションの基本設定などが含まれる HKU¥<UserName>キーの別名
HKEY_LOCAL_MACHINE (HKLM)	OS，ハードウェアに関する情報を保存するレジストリ システムメモリ容量，デバイスドライバ情報，Windowsサービス情報などが含まれる
HKEY_USERS (HKU)	全ユーザーの情報(プロファイル)を保存するレジストリ HKEY_USERSのうち，現在ログオンしているユーザー部分が，別名でHKEY_CURRENT_USERとしてアクセスされる
HKEY_CURRENT_CONFIG (HKCC)	コンピュータが現在システム起動に使用しているハードウェア情報を保存するレジストリ HKLM¥SYSTEM¥CurrentControlSet¥HardwareProfiles¥Currentキーの別名

●アプリケーションが使用するレジストリ

一般のデスクトップアプリケーションでは，画面上の表示位置などをレジストリに保存し，アプリケーションが再度起動された時にレジストリから読み込み，表示位置を制御するなどの処理をおこなっています．デスクトップアプリケーションが使用するレジストリは，現在ログオンしているユーザー用のレジストリ「HKEY_CURRENT_USER」です．Visual Basic 6.0以前のバージョンからサポートされているGetSettings，SaveSettingsでは，「HKEY_CURRENT_USER」に対してしかアクセスできません．それ以外のルートキー配下のレジストリにアクセスするためには，WIN32APIを使用する必要がありました．

サーバーアプリケーションなどのバックグランドで動作するアプリケーションでは，ユーザーがログオンしていない状態でも動作しますので，ローカルコンピュータのレジストリ「HKEY_LOCAL_MACHINE」を使用します．

●レジストリアクセス

レジストリからの読み取り，レジストリへの書き込みは，「Microsoft.Win32.Registryクラスを使用する方法」と，「WMI System Registryプロバイダが提供するクラスを使用する方法」があります．前者はWMIを使用しない方法で，後者がWMIを使用した方法です．

なお，Win32プロバイダが提供するWin32クラスのオペレーティングシステムに含まれるWin32_Registryクラスは，レジストリの管理情報（サイズ，位置など）を表すクラスです．第2章で使用したWMIプロバイダは「Win32プロバイダ」で，レジストリアクセスに使用するのは「System Registryプロバイダ」です．

3.1.2 レジストリアクセス（Registryクラス）

本書はWMIプログラミングに関する書籍ですが，参考のため.NET FrameworkのMicrosoft.Win32.Registryクラスを使用したレジストリアクセスについても説明します．

●Registryクラス

.NET FrameworkのMicrosoft.Win32.Registryクラスを使用すれば，他の.NET Frameworkクラスライブラリと同様な方法でレジストリにアクセスすることができます．Registryクラスの使用方法は，LocalMachineフィールドのOpenSubKeyメソッドでレジストリをオープンし，GetValueメソッドでレジストリの値を取得（または，SetValueメソッドでレジストリの値を設定）し，Closeメソッドでレジストリへのアクセスを終了する流れとなります．

●レジストリからの読み取り

Registryクラスを使用した，レジストリからの読み取りの例（レジストリのオープン・リード・クローズ）を以下に示します．この例では，「HKEY_LOCALMACHINE」配下の「SYSTEM¥CurrentControlSet¥Services¥Service1¥Cycle」パラメータの値を取得しています．

```
Imports Microsoft.Win32
  :
Dim regKey As RegistryKey
Dim strReg As String
Dim lngReg As Long

' レジストリのオープン
regKey = Registry.LocalMachine.OpenSubKey( _
            "SYSTEM¥CurrentControlSet¥Services¥Service1", False)

' レジストリの読み取り
strReg = "Cycle"
lngReg = regKey.GetValue(strReg)

' レジストリのクローズ
regKey.Close()
```

●レジストリへの書き込み

Registryクラスを使用した，レジストリへの書き込みの例(レジストリのオープン・ライト・クローズ)を，以下に示します．この例では，「HKEY_LOCALMACHINE」配下の「SYSTEM¥CurrentControlSet¥Services¥Service1¥Cycle」パラメータに値(1000)を設定しています．

```
Imports Microsoft.Win32
  :
Dim regKey As RegistryKey
Dim strReg As String
Dim lngReg As Long

' レジストリのオープン
regKey = Registry.LocalMachine.OpenSubKey( _
            "SYSTEM¥CurrentControlSet¥Services¥Service1", True)

' レジストリの書き込み
strReg = "Cycle"
lngReg = 1000
regKey.SetValue(strReg, lngReg)

' レジストリのクローズ
regKey.Close()
```

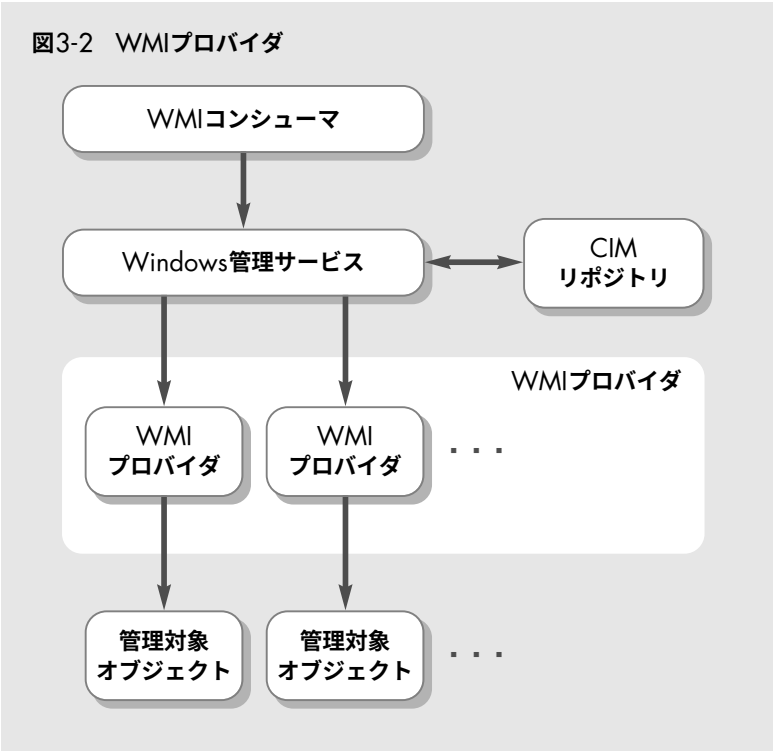
3.1.3 レジストリアクセス(WMI System Registry)

ここでは，WMIプロバイダ・CIMリポジトリ，WMIプロバイダのSystem Registryプロバイダを使用したレジストリアクセスについて説明します．

●WMIプロバイダ

「1.2 WMIの概要」でも説明しましたが，WMIプロバイダは，Windows管理サービスから呼び出され，各種のWMIプロバイダが管理する管理対象オブジェクトから情報収集，情報設定をおこなうDLLです．(**図3-2**）第2章で使用したWMIプロバイダは「Win32プロバイダ」で，本章でレジストリアクセスに使用するのは「System Registryプロバイダ」です．

図3-2 WMIプロバイダ



●CIMリポジトリ

CIMリポジトリは，システム管理情報にアクセスするためのクラスやオブジェクトデータを管理するデータベースで，「WinMgmt.EXE」という名称のWindowsシステムサービスの一部として実装され，COMインターフェイスを経由してアクセスされます．COMインターフェイスですので，レジストリのHKEY_CLASSES_ROOTにCOMの持つCLSIDと実際に動作するCOM DLLの名称が登録されています．Win32プロバイダ，System RegistryプロバイダのCLSIDとCOM DLL名称を，**表3-4**に示します．

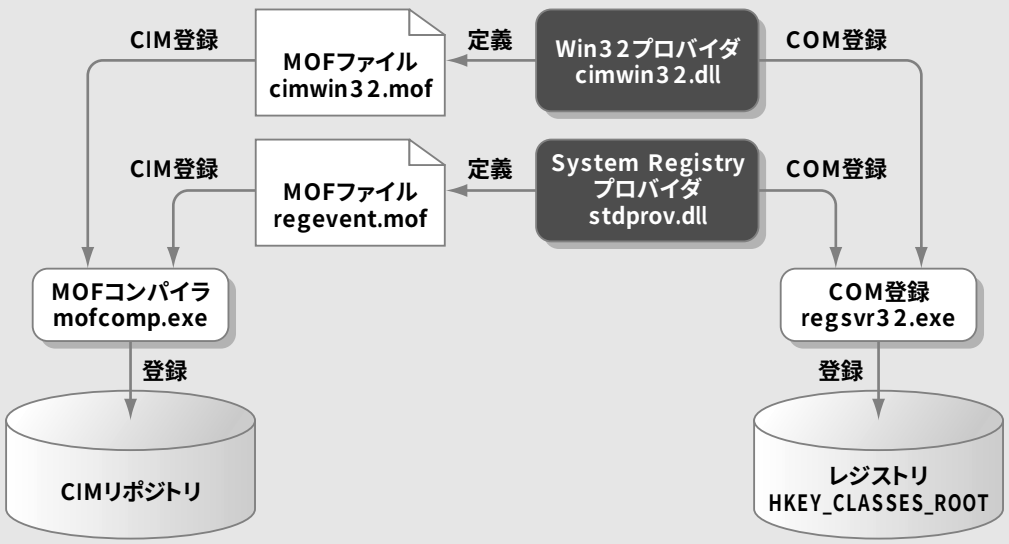
表3-4　WMIプロバイダ情報

WMIプロバイダ名	CLSID／COM DLL名称
Win32プロバイダ	{d63a5850-8f16-11cf-9f47-00aa00bf345c} cimwin32.dll
System Registryプロバイダ	{fe9af5c0-d3b6-11ce-a5b6-00aa00680c3f} stdprov.dll

●MOFファイル

MOF(Managed Object Format)ファイルは，CIMリポジトリにシステム管理情報にアクセスするためのクラスやオブジェクトデータを登録するためのテキスト形式ファイルです．MOFファイルは，Windows 2000/XPの場合，システムディレクトリ(WINNT¥System32またはWINDOWS¥System32)配下の「WBEM¥Repository」ディレクトリに実装されています．また，Microsoftが標準で提供するMOFファイルも，システムディレクトリ配下の「WBEM」ディレクトリに存在します．Win32プロバイダ，System Registryプロバイダともに，標準でMOFファイルが提供され，CIMリポジトリに情報が登録されていますので，MOFファイルの登録などの操作は必要ありません．MOFファイルは，MOFコンパイラ(mofcomp.exe)でコンパイルし，CIMリポジトリに登録します．MOFファイルとWMIプロバイダ(COM DLL)の関連を，**図3-3**に示します．

図3-3　MOFファイルとWMIプロバイダ



なお，独自のWMIプロバイダを開発・利用するためには，WMIプロバイダをCOM DLLとしてレジストリに登録し，WMIプロバイダに対応したMOFファイルを作成・コンパイルしてCIMリポジトリに登録する必要があります．

●System Registryプロバイダが提供するクラス

System Registryプロバイダが提供するレジストリアクセスに使用クラスは，「StdRegProvクラス」だけです．StdRegProvクラスは，レジストリにアクセスするためのメソッドはありますが，プロパティはありません．StdRegProvクラスのメソッドを，表3-5に示します．

表3-5　StdRegProvクラスのメソッド

メソッド名	説　明
CreateKey	サブキーを作成
DeleteKey	サブキーを削除
EnumKey	サブキーを列挙
EnumValues	キーの値を列挙
DeleteValue	キーの値を削除
SetDWORDValue	REG_DWORD型データを設定
GetDWORDValue	REG_DWORD型データを取得
SetStringValue	REG_SZ型データを設定
GetStringValue	REG_SZ型データを取得
SetMultiStringValue	REG_MULTI_SZ型データを取得
GetMultiStringValue	REG_MULTI_SZ型データを取得
SetExpandedStringValue	REG_EXPAND_SZ型データを取得
GetExpandedStringValue	REG_EXPAND_SZ型データを取得
SetBinaryValue	REG_BINARY型データを設定
GetBinaryValue	REG_BINARY型データを取得
CheckAccess	ユーザープロセスが指定されたアクセス権を所有しているかをチェック

●レジストリアクセスの例

System Registryプロバイダが提供するStdRegProvクラスを使用した，レジストリからの読み取り例を以下に示します．この例では，HKEY_LOCAL_MACHINEのSYSTEM¥CurrentControlSet¥Services¥Tcpip¥Typeの値を取得しています．

```
' Scope, Path, Optionを指定して，ManagementClassを作成
Dim objScope As New ManagementScope("¥root¥default")
Dim path As New ManagementPath("StdRegProv")
Dim opt As New ObjectGetOptions(Nothing)
Dim regClass As New ManagementClass(objScope, path, opt)

' メソッド(GetDWORDValue)を呼び出す際の入力パラメータを取得
Dim inParams As ManagementBaseObject = _
```