Using the WMI Adapter for CIMPLE

Michael E. Brasher
November 4, 2008

Table of Contents

1 Introduction

This document explains how to use the WMI adapter for CIMPLE. The WMI Adapter enables CIMPLE providers to work with the Windows Management Instrumentation (WMI) server. The following chapters explain how to:

- install CIMPLE for WMI
- develop a trivial provider
- register the WMI provider
- verify the provider

Except for a few minor details, provider developement for WMI is similar to developing providers for other servers. After reading this guide, you will be able to make existing CIMPLE providers work with WMI.

2 Installing CIMPLE for WMI

To build CIMPLE and the WMI adapter, open a Windows terminal session (`cmd.exe`). Run the Visual Studio setup scripts. Verify that the Visual Studio C++ compiler is on your path (`cl.exe`). Obtain GNU make (`make.exe`) and add it to your path. GNU make for Windows is available from the "tools" link on the OpenPegasus home page (see http://openpegasus.org).

Unpack the CIMPLE distribution. From the root of the distribution type.

```
C:\> configure.bat --bindir=c:/windows/system32 --enable-wmi
```

Next build CIMPLE by typing:

```
C:\> make
```

Finally, install CIMPLE as follows:

```
C:\> make install
```

This installs the programs and DLLs under:

```
C:\windows\system32
```

All other CIMPLE files are installed under:

```
C:\cimple
```

You may uninstall CIMPLE later by typing:

```
C:\> make uninstall
```

3 Building a trivial provider

This chapter explains how to build a trivial provider. For the most part, it is like building a CIMPLE provider for other servers, but there are a few minor differences.

We start with the following MOF definition (which we place in `repository.mof`).

```
[dynamic, provider("Person")]
class Person
{
    [Key] string SSN;
    [Key] string FirstName;
    [Key] string LastName;

    [implemented]
    uint32 foo([in] string arg);
};
```

The `dynamic`, `provider`, and `implemented` qualifiers are unique to WMI. The string given by the `provider` qualifier must match the name of the provider DLL (without the extension).

Next we use the `genprov` command to generate the classes, provider, and module.

```
C:\> genmod Person Person
Created Person.h
Created Person.cpp
created repository.h
Created repository.cpp
Created Person_Provider.h
Created Person_Provider.cpp
Created module.cpp
Created guid.h
Created register.mof
```

This generates the following files:

- `Person.h` - the Person class declaration
- `Person.cpp` - the Person class definition
- `repository.h` - the class repository declarations
- `repository.cpp` - the class repository definitions
- `Person_Provider.h` - the Person provider declaration
- `Person_Provider.cpp` - the Person provider methods
- `module.cpp` - the WMI entry points.
- `guid.h` - the GUID that uniquely identifies the provider COM server.
- `register.mof` - the WMI registration instances.

Next we must create a `link.def` file shown below.

```
LIBRARY "Person.dll"

EXPORTS
        DllMain PRIVATE
```

```
        DllCanUnloadNow PRIVATE
        DllGetClassObject PRIVATE
        DllRegisterServer PRIVATE
        DllUnregisterServer PRIVATE
```

Then we create the following Makefile.

```
TOP=../../..
include $(TOP)/mak/config.mak

LIBRARY = Person
SOURCES = Person.cpp Person_Provider.cpp module.cpp repository.cpp
LIBRARIES = cimplewmiadap cimple
EXTRA_LINK_FLAGS = /def:link.def
EXTRA_SYS_LIBS = ole32.lib oleaut32.lib
DEFINES += -DCIMPLE_WMI_MODULE

include $(TOP)/mak/rules.mak
```

Finally, we build the provider as shown below.

```
C:\> make
```

This creates a DLL called `Person.dll`.

4 Registering a WMI provider

This chapter shows how to register a WMI provider. First we must copy the provider DLL to the WMI providers directory, Usually located here:

```
C:\windows\system32\wbem\
```

Second we use the WMI MOF compiler to add our classes to the CIM repository as shown below.

```
mofcomp repository.mof
```

Third we register our provider as follows.

```
mofcomp register.mof
```

Finally, we register our WMI provider as a COM server:

```
regsvr32 /s C:\windows\system32\wbem\Person.dll
```

5 Verifying a WMI provider

We recommend using the WMI `wbemtest.exe` client program to verify providers you write with CIMPLE. You may also use many other WMI client tools. For a full discussion of these, see the book "Developing WMI solutions: A Guide to Windows Management Instrumentation" by Craig Tunstall and Gwyn Cole.