CIMPLE 1.2.0 Release Notes

Michael E. Brasher
Jan 15, 2008

Table of Contents

1 Introduction

This document introduces CIMPLE 1.2.0 and explains what has changed since the last public release (CIMPLE 1.1.0). If you are unfamiliar with the major changes introduced by CIMPLE 1.1.0, you might want to review them by clicking here: http://www.cimple.org/downloads.html.

Chapter 1 describes what is new in this release. Chapter 2 discusses bugs fixed by this release. Chapter 3 explains how to migrate providers developed with earlier versions.

2 What's New?

This chapter covers new capabilities introduced by CIMPLE 1.2.0.

2.1 BREVITY

BREVITY is a new client interface that employs concrete CIM elements (classes and methods). BREVITY is described in a presentation presented at the 2007 Management Developers Conference. For more information, please see http://www.cimple.org/CIMPLE-2007-MDC.pdf.

2.2 Platform Support

CIMPLE 1.2.0 now supports the following platforms.

- Linux-X86 32-bit, GNU C++
- Linux-X86 64-bit, GNU C++
- Linux-IA64 64-bit, GNU C++
- Linux-S390 32-bit, GNU C++
- Linux-S390 64-bit, GNU C++
- Linux-PPC 32-bit, GNU C++
- Linux-PPC 64-bit, GNU C++
- Darwin-X86-32-bit, GNU C++
- Solaris-SPARC-64-bit, GNU C++
- VxWorks-XScale-32-bit, GNU C++
- Windows-X86-32-bit, MSVC
- Windows-X86-64-bit, MSVC

The following were added in this realease.

- Darwin-X86-32-bit, GNU C++
- Solaris-SPARC-64-bit, GNU C++
- VxWorks-XScale-32-bit, GNU C++

2.3 New `genmak` Tool

The new `genmak` tool generates provider makefiles automatically (the CIMPLE build system is now installed by the install target). The following generates a makefile that builds a library named "xyz" from a set of sources.

```
$ genmak xyz *.cpp
```

The `genproj` command (which runs genclass, genprov, and genmod) also runs genmak, when passed the `-m` option.

Developers may continue to write their own makefiles if they prefer. This tool is simply a convenience that simplifies provider development.

2.4 Logging Facility

This version introduces a logging facility that may be used in the provider. Logging is only enabled when CIMPLE is configured with debug. For example:

```
$ ./configure --enable-debug ...
```

All log entries are written to:

```
$HOME/.cimple/messages
```

Provider use the logging by calling the `log` function. For example:

```
log(LL_DBG, "my name is %s; my age is %d", "John", 12);
```

There are five log levels:

- `LL_FATAL`
- `LL_ERR`
- `LL_WARN`
- `LL_INFO`
- `LL_DBG`

In addition to provider generated log entries, the CIMPLE adapters themselves make log entries for warnings and errors during the normal operation of CIMPLE. We recommend configuring with debug and watching for log messages during provider development.

3 Bug Fixes

This release fixes the bugs described below (all critical bugs were addressed by earlier maintenance releases).

3.1 CMPI Header Compliation Errors

This release works around errors in the the CMPI headers released last year on the Open-Group site. CIMPLE now compiles with all known CMPI headers.

4 Migration Notes

Always regenerate classes, providers, and modules when using a new version of CIMPLE. This is a trivial matter of running `genrpoj` as follows:

```
$ genproj MODULE-NAME CLASS-1 CLASS-2 ... CLASS-N
```

This regenerates the classes and module and will patch your providers if necessary. This operation will not require any rework on your part. Just regnerate, clean, and remake.