Tracing Framework

User Guide

1. Tracing Framework: User Guide

The CIMOM provides a tracing facility that helps in investigating the cause of a problem. For example, if requests or Providers abort, performance is reduced, or unexpected responses appear, trace file can provide pointers to where and when the problem occurred.

1.1 Steps for activating tracing

1. The tracing of CIMOM can be configured by setting the following properties.
   - traceLevel
   - traceFilePath
   - traceComponent

   Refer to the CIMOM Configuration framework document for more information regarding setting/unsetting of the CIMOM configuration properties using the cimconfig command. (cimconfig.html man page is located in the directory, <PEGASUS_ROOT>/src/Clients/cimconfig/doc/).

   The changes to the trace properties will take into effect immediately. Hence, there is no need to re-start CIMOM.

2. Set the required trace level. The trace level indicates the level of information to be included in the trace output. The following are the valid trace levels.

Level #	Description
Level 1	Function entry/exit.
Level 2	Basic logic flow trace messages, minimal data detail
Level 3	Intra function logic flow and moderate data detail
Level 4	High data detail

   Each successive level provides more detailed information and includes information from the levels above it. The default trace level is set to “1”. This will enable function entry/exit trace messages.

3. Set the trace file. The trace file will contain the trace output information. By default the trace file, cimom.trace is created in the {$PEGASUS_HOME}/logs directory.

4. Add the components to be traced. The list available trace components is defined in Section 1.3. If you need to trace all the components specify “ALL”. None of the components are enabled by default.

5. View the trace output file.

1.1.1 Examples

The following examples use the cimconfig command to update the current value of the trace properties.

Set the trace component to ALL. This will enable tracing for all available components.

```
$cimconfig –s traceComponents=ALL -c
```

Set trace level to 1

```
$cimconfig –s traceLevel=1 -c
```

Set the trace file

```
$cimconfig –s traceFilePath=/tmp/cimom.trace -c
```

The following examples use the cimconfig command to view the trace properties.

View the property, traceComponents

```
$cimconfig –g traceComponents -c
```

View the property, traceLevel

```
$cimconfig –g traceLevel=1 -c
```

View the property, traceFilePath

```
$cimconfig –g traceFilePath=/tmp/cimom.trace -c
```

1.2 Interpreting the trace output file

The following is the standard trace output format:
<*Timestamp*>:<*Component Name*> <*File name: Line Number*> : <*detailed information*>

The following example shows a sample trace output file:
11:53:18-07/30/01: XML_Reader [XmlReader.cpp:249]: Entering method [XmlReader:: testCimStartTag]
11:53:18-07/30/01: XML_ Reader [XmlReader.cpp:270]: Exiting method [XmlReader:: testCimStartTag]

1.3 List of Trace Components

Component	**Component Name**
Channels	Channel
XML Parser	XmlParser
XML Writer	XmlWriter
XML Reader	XmlReader
HTTP Protocol	Http
CIM Types, Value, Objects	CimData
Provider Manager	ProvManager
Repository	Repository
Request dispatching	Dispatcher
OS Abstractions	OsAbstraction
Configuration	Config
Indication Delivery	IndDelivery