

## Contents

- [1 Pod Tracing](#)
- [2 Unified System CLI](#)
- [3 EIM/WIM Trace Levels](#)
- [4 How the trace levels correlate to the log files](#)
- [5 Trace Level Recommendations](#)
- [6 Disk Space Concerns](#)
- [7 egpl\\_log4j.xml Corruption](#)
- [8 Diving in](#)
- [9 Setting Trace Levels](#)

## Pod Tracing

All pods have had the following tracing pre-configured.

Component	Tracelevel
EIM/WIM processes	INFO
MR PIM	<pre>procmon &lt;cust inst&gt; PGxx pimx trace mr* /on trace mr_function_call /off trace mr_heartbeat_messages /off</pre>
CTI Server	0xF8
OPC	<pre>opctest /cust &lt;cust_inst&gt; /node PGxx debug /agent /closedcalls /ctimsg /routing /cstacer /inrcmsg /tpmsg /calls /realtime /task EMSTraceMask 0x40</pre>
Router	<pre>\icm\bin\rttrace.exe RouteRequests, Translation Route, NetworkVRUTracing, CallTypeRealtime, Agent Changes</pre>
CMB	Verbose Level 10

## Unified System CLI

Starting in 8.x, we can avoid the registry and use the CLI to set tracing for UCCE components with a simple "debug level 1 ...".

## Tracing

```
Unified System CLI
Enter username [DSLAB\Administrator]:
Enter password:
Enter Instance [ucel(Optional)]:

Welcome to the Platform Command Line Interface

admin:system
Initializing system mode ...
Retrieving device list. This process may take a few minutes to complete.
*****
admin(system):debug level 1 ?
Options: component
         devicetype
         dtcomponent
         dtsubcomponent
         group
         redirect
         server
         subcomponent
         sysmatch
         <cr>

admin(system):_
```

These same trace levels can be set from the web-based Portico Diagnostic Framework.

The screenshot displays the Portico Diagnostic Framework web interface. On the left, a 'Commands' menu lists various categories: Alarm, Configuration, Inventory, License, Log, Network, Performance, Platform, Service, and Trace. The 'Trace' category includes 'SetTraceLevel', which is highlighted with a red box. A red arrow points from this box to the 'SetTraceLevel' configuration form on the right. The form includes a 'Component' dropdown set to 'Router A/mds', a 'Level' dropdown set to '0', and checkboxes for 'TraceSettingCookie' and 'Show URL'. A 'Submit' button is located below these options. A text box on the right states: 'Components will always be local to the server from where you are running the Diagnostic Framework Portico'.

## Tracing

Since UCCE does not use OAMP like CVP, there is some minor configuration required. Reference the [UCCE Serviceability Best Practices Guide](#) for details on how to configure your devices.csv file.

### EIM/WIM Trace Levels

Starting with EIM/WIM 4.3(2), there are 8 different levels of logging.

1. Fatal
2. Error
3. Warn
4. Info
5. Perf
6. DBquery
7. Debug
8. Trace

### How the trace levels correlate to the log files

Each line of EIM/WIM process logs tell you which level it comes from. Below, we see messages from 3 different levels: WARN, ERROR, and INFO. Higher levels will include messages from lower levels.

```
2011-05-01 19:02:23.758 GMT-0400 <@> WARN <@> [32:Thread-6] <@> ProcessId:1740 <@> PID:1 <@> UID:1
2011-05-01 19:02:23.774 GMT-0400 <@> ERROR <@> [32:Thread-6] <@> ProcessId:1740 <@> PID:1 <@> UID:1
2011-05-01 19:02:23.774 GMT-0400 <@> INFO <@> [32:Thread-6] <@> ProcessId:1740 <@> PID:1 <@> UID:1
```

### Trace Level Recommendations

The default trace level for EIM/WIM components is ERROR. In most cases, ERROR level logs are not sufficient to find root cause.

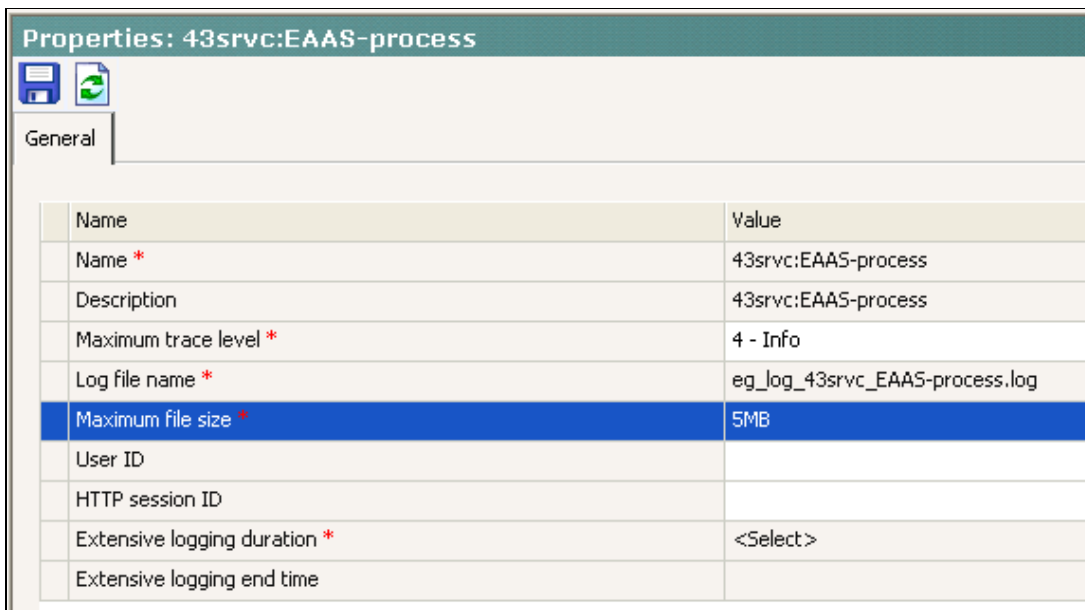
DEBUG gives a significant bit more detail (i.e. SQL queries being run) that can be useful for low-level investigations, but is in many ways redundant and not absolutely necessary for troubleshooting. Starting with 4.3(2), DEBUG level logging will automatically return to ERROR level after a user-specified period of time. INFO level logging will not do this.

This entire class will use INFO level tracing for EIM/WIM components. It is a safe trace level for production usage as long as Virtual Memory is set correctly per the 4.3 Deployments Guide: *Virtual memory setting should be set to 1.5 times the physical memory. To ensure that adequate space is available during run time, distribute the virtual memory across disk volumes. If multiple processes are set to INFO level without adequate virtual memory allocated, the system will start to crawl and eventually halt altogether.*

### Disk Space Concerns

Note that log file size is constant at 5MB. It is not possible to change this from the System Console > Logger folder in the UI.

## Tracing



Name	Value
Name *	43srcv:EAA5-process
Description	43srcv:EAA5-process
Maximum trace level *	4 - Info
Log file name *	eg_log_43srcv_EAA5-process.log
Maximum file size *	5MB
User ID	
HTTP session ID	
Extensive logging duration *	<Select>
Extensive logging end time	

If the logging level of a process is increased past ERROR, a **total of 100x 5MB log files will be written for the process**. EIM will automatically overwrite the oldest files once this limit is reached.

- The size of each file is set in C:\CIMeService\config\egpl\_log4j.xml.
- This also tells processes what to name their logfiles, and where to save them (File Server).

```
<appender class="com.egain.platform.util.logging.appender.RollingFileAppender" name="eimServices:
<param name="Threshold" value="info#com.egain.platform.util.logging.Level"/>
<param name="File" value="eg_log_eimServices_rules-cache-process.log"/>
<param name="MaxBackupIndex" value="-1"/>
<param name="MaxFileSize" value="5MB"/>
```

## egpl\_log4j.xml Corruption

The egpl\_log4j.xml file can sometimes become corrupted, causing process startup and logging to fail. If only one or two java/javaw processes are starting, this file is likely corrupted.

- A clean version of this file is included in the same directory as "egpl\_log4j.pristine"
- In the event of file corruption, the pristine file can be copied and overwrite the current egpl\_log4j.xml.
- Start Services with the fresh file, and EIM/WIM will "finish" the file's contents much like it does with a JBoss directory when you first start a newly-created Application Server.

## Diving in

In ICM/UCCE logs, a common first step is to grep for strings such as "Exception" or "Error". With EIM/WIM logging, this is not useful due to the sheer amount of results returned.

- All ERROR-level log lines will be returned, even though they may not actually be errors.
- There are a large number of exceptions occurring throughout the system even in steady-state behavior. Many of these are benign or unrelated to whatever issue you may be troubleshooting. The best option is to start with something specific to your issue - an activity ID, case ID, user ID, queue ID, etc.
- **Note that Activity IDs will belong to a Case ID.** This will typically be a many-to-one relationship.

## Setting Trace Levels

Tracing is set from the System partition: <http://<webserver>/system>

*View a [video example](#) of setting trace levels.*