Cisco Prime Network Registrar IPAM 8.3 Logging Overview

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Cisco Prime Network Registrar IPAM 8.3 Logging Overview

Contents

This page intentionally left blank.

Introduction

This document summarizes the logs that Cisco Prime Network Registrar IPAM 8.3. provides. Generally each service or daemon provided with IPAM provides one or more logging facilities. This document highlights these services, default logging locations and if appropriate, configuration of logging functionality. In all log pathnames, $INCHOME refers to the home directory of IPAM on the server in question, /opt/incontrol on Linux/Unix systems and C:\Program Files\Cisco\Prime Network Registrar IPAM on Windows systems by default.

Most but not all services utilize the log4j java logging utility. Log4j provides a convenient means of managing and changing logging levels for IPAM services and supports the following logging levels from highest/coarsest to lowest/finest:

- FATAL
- ERROR
- WARN
- INFO
- DEBUG

We'll review specific configuration examples as we discuss each service.

Note: CPNR IPAM 8.3 and later versions will not support Solaris. Refer to earlier versions of IPAM documents if you want to use IPAM with Solaris support.

IPAM Executive Services

The IPAM Executive consists of the Tomcat web server, a MySQL database, and various IPAM Executive services as described in this section.

Tomcat Web Server

Tomcat provides logging for access and execution of Tomcat processes and IPAM business logic. Tomcat logging is configured via a variety of log4j properties files and xml configuration files as described below.

InControl Log – The InControl log logs events for the IPAM application and business logic execution to $INCHOME/tomcat/logs/incontrol.log by default. This logging output destination is defined in the log4j.properites file in the $INCHOME/tomcat/webapps/incontrol/WEB-INF/classes directory per the following setting:

```
log4j.appender.A2.File=${catalina.base}/logs/incontrol.log
```

The ${catalina.base} directory is the home directory for Tomcat within the IPAM installation which is $INCHOME/tomcat. Other incontrol.log file parameters that can be set in this properties file include the maximum file size, which defaults to 25MB (`log4j.appender.A2.MaxFileSize=25MB`) and number of rollover versions

(`log4j.appender.A2.MaxBackupIndex=4`). Otherwise, the severity of events logged can be defined by log category; for example:

```
log4j.category.com.diamondip=DEBUG
log4j.category.com.diamondip.common=INFO
log4j.category.org.hibernate.SQL=ERROR
log4j.category.org.hibernate.type=ERROR
log4j.category.com.opensymphony.xwork.interceptor=ERROR
log4j.category.com.opensymphony.xwork.ActionSupport=ERROR
```

IPAM Web Service Log – The web service logs events for IPAM web services (inc-ws) access and business logic to `$INCHOME/tomcat/logs/webservice.log` by default. This logging output destination is defined in the log4j.properites file in the $INCHOME/tomcat/webapps/inc-ws/WEB-INF/classes directory per the following setting:

```
log4j.appender.A2.File=${catalina.base}/logs/webservice.log
```

The logging category event severity setting can be edited on the lines:

```
log4j.category.com.diamondip=DEBUG

log4j.category.org.hibernate.SQL=DEBUG
```

Tomcat Localhost log - - The `$INCHOME/tomcat/conf/server.xml` file contains the directory specification and file-naming format for the daily tomcat localhost log. The default path and file format is $INCHOME/tomcat/logs/localhost_log.<date>.txt where <date> is the current date in YYYY-MM-DD format. This is specified by the directory=”logs” parameter shown below, which enables entry of an absolute pathname or a subdirectory of ${catalina.base} (i.e., $INCHOME/tomcat).

```
<Logger className="org.apache.catalina.logger.FileLogger"
    directory="logs"  prefix="localhost_log." suffix=".txt"
    timestamp="true"/>
```

The format of the log file is also specified in this entry by the prefix fixed text (localhost_log.), suffix (.txt) and use of timestamp in between.

Tomcat Localhost administrator log - The `$INCHOME/tomcat/conf/Catalina/localhost/admin.xml` file contains the directory specification and file-naming format for daily tomcat localhost administrator log, which logs Tomcat configuration changes. The default path and file format is `$INCHOME/tomcat/logs/localhost_admin_log.<date>.txt` where <date> is the current date in YYYY-MM-DD format. This is specified by the directory=”logs” parameter shown below, which enables entry of an absolute pathname or a subdirectory of ${catalina.base} (i.e., $INCHOME/tomcat).

```
<Logger className="org.apache.catalina.logger.FileLogger"
   directory="logs"  prefix="localhost_admin_log." suffix=".txt"
   timestamp="true"/>
```

The format of the log file is also specified in this entry by the prefix fixed text (localhost_admin_log.), suffix (.txt) and use of timestamp in between.

Velocity template engine logging – The velocity log is specified via the `runtime.log = velocity.log` (relative to the server's tomcat (CATALINA) home directory, $INCHOME/tomcat) statement in the velocity.properties file in the $INCHOME/tomcat/work/Catalina/localhost/incontrol/loader/com/diamondip/common/filemerge/ directory.

Web Services – Command line interfaces (CLIs) and API calls utilize web services to enable direct or scripted transaction interfaces to IPAM. The default log file for web such web services is the $INCHOME/etc/cli/ns_apache_webservice.log file. This is configured in the $INCHOME/etc/cli/log4j.xml file via the statement block:

```
   <appender name="FILE_apache"
class="org.apache.log4j.RollingFileAppender">
      <param name="File" value="log/ns_apache_webservice.log"/>
      <param name="Append" value="true" />
      <param name="MaxFileSize" value="1000KB" />
      <param name="MaxBackupIndex" value="2" />
      . . .
   </appender>
```

The file name is relative to the current directory, yielding the output file as: $INCHOME/etc/cli/log/ns_apache_webservice.log. Similar configuration of ns_websevice.log is enabled in the same log4j.xml file. Association of logging categories and corresponding severity is also specified in this file, using statements such as:

```
      <logger name="com.diamondip.ipcontrol" additivity="true">
        <level value="error" />
        <appender-ref ref="FILE_apache" />
      </logger>
```

The ImportDNS API/CLI has its own logging facility. This is configured in the `dns_import_log4j.properties file` in the $INCHOME/etc/cli directory. The default logging output file location is $INCHOME/etc/cli/log/dnsimport.log. Categories and corresponding event severities to log may be defined in the statements at the end of the file:

```
      log4j.category.com.diamondip.ipcontrol.cli.dnsimport=INFO
      log4j.category.com.diamondip.netcontrol=WARN
      log4j.category.com.diamondip.ipcontrol=WARN
```

MySQL Database

MySQL logging must be enabled by modifying the mysqld_start script. One or more of the following clauses corresponding to chosen logging functionality should be inserted into the ./bin/mysqld_safe line of $INCHOME/etc/mysqld_start.

Query Log – This log provides a general record of MySQL activites, such as client connection or disconnection, as well as each SQL statement received from clients. The following clause should be added to the mysql_start script per above to log to the $INCHOME/log/mysql-queries.log file:

```
--log=${INCHOME}/log/mysql-queries.log
```

Error Log – This log includes information indicating when mysqld was started, restarted, and stopped, as well as any critical errors that occurred while mysqld was running. The following clause should be added to the mysql_start script per above to log to the $INCHOME/log/mysql-error.log file:

```
--log-error=${INCHOME}/log/mysql-error.log
```

Binary Log – The binary log contains all statements that update data or potentially could have updated data (e.g., an update query with no matching records). This does not include queries that do not update data (show or select).

```
--log-bin=${INCHOME}/log/base_name
```

Slow Query Log – This log records queries which took longer to execute than the value of the `long_query_time` variable, which defaults to 10 seconds.

```
--log-slow-queries=${INCHOME}/log/mysql-slowqueries.log
```

IPAM Executive Services

The following services run on the Executive with corresponding logging destinations. Most of these services utilize the log4j java logging utility. The log levels for output can be set per service by editing the corresponding <service>_log4j.properties file in $INCHOME as discussed below for each service, though the corresponding service will need to be restarted to enact any changes.

Task Manager – The Task Manager uses log4j and logs to $INCHOME/log/taskmgr.log by default. This output destination can be changed by editing the $INCHOME/task_manager_log4j.properities file by editing the line:

```
log4j.appender.RollingFile.File=${INCX_HOME}/log/taskmgr.log
```

Other parameters such as the maximum file size and number of file backups can also be edited if necessary. The output logging level can be set by modifying the following statement, shown as set to corresponding defaults:

```
log4j.category.com.diamondip.ipcontrol=INFO
log4j.category.com.diamondip.netcontrol=INFO
log4j.category.com.diamondip.ipcontrol.task=DEBUG
log4j.category.com.diamondip.netcontrol.task=DEBUG
log4j.category.com.diamondip.netcontrol.taskmgr=DEBUG
```

Setting the log level configures logging of event of the specified severity and higher; e.g., setting this to INFO will inhibit output of DEBUG events but will include INFO, WARN, ERROR and FATAL events.

Result Manager – You may have one or two Result Manager service versions installed on your Executive. If you are supporting remote agents that are IPAM 8.0 and above, you should be running only the Result Manager daemon (not v2).

The output destination for Result Manager is $INCHOME/log/resultmgr.log and can be set by editing the $INCHOME/result_manager_log4j.properities file by editing the line:

```
log4j.appender.RollingFile.File=${INCX_HOME}/log/resultmgr.log
```

The logging level can be set by editing the following line in either file (the default levels as shown):

```
log4j.category.com.diamondip.ipcontrol=INFO
log4j.category.com.diamondip.netcontrol=INFO
log4j.category.com.diamondip.ipcontrol.task=DEBUG
log4j.category.com.diamondip.netcontrol.task=DEBUG
log4j.category.com.diamondip.netcontrol.resultmgr=DEBUG
```

Callout Manager – The Callout Manager also uses log4j and logs Callout Manager state changes, events, and callout activity including detailed callout script file contents to $INCHOME/log/calloutmgr.log by default. This and other Callout Manager logging properties can be changed by editing the callout_manager_log4j.properties file in the $INCHOME directory, particularly the

```
log4j.appender.RollingFile.File=${INCX_HOME}/log/calloutmgr.log
```

output destination specification and the logging level, DEBUG by default:

```
log4j.category.com.diamondip.netcontrol=DEBUG
log4j.category.com.diamondip.netcontrol.calloutmgr=DEBUG
```

DNS Listener – The DNS Listener service uses log4j as well and logs to $INCHOME/log/dnslistener.log by default. This output destination can be modified by editing the following gentry in the $INCHOME/dns_listener_log4j.properties file:

```
log4j.appender.RollingFile.File=${INCX_HOME}/log/dnslistener.log
```

The logging level, INFO by default, can be modified via the following entry in this same properties file:

```
log4j.category.com.diamondip.ipcontrol.dnssyncmgr=INFO
log4j.category.com.diamondip.netcontrol=WARN
log4j.category.com.diamondip.ipcontrol=WARN
```

File Manager – The File Manager does NOT use log4j, but is configured, including its logging configuration via the $INCHOME/ftpd.conf file. The logging output destination is set by default to $INCHOME/log/ftp.log. The logging output destination is specified via the `FtpServer.server.config.data=${INC_HOME} line`. This parameter is overloaded as it also specifies other File Manager resource file locations; the File Manager will log output to an ftp.log file within a log subdirectory of the specified path; hence the output to $INCHOME/log/ftp.log.

The File Manager supports four logging levels, from coarsest to finest: Error (3), Warning (2), Information (1), and Debug(0). The default setting is Information (1) but this can be changed by editing the `FtpServer.server.config.log.level=0` line, shown set to Debug (=0).

Log Manager – The Log Manager was intended to provide a cross-service logging perspective utilizing the log4j utility. The default logging output destination was $INCHOME/log/logmanager.log as defined in the lm_log4jconfig.properites file, though currently logging is disabled.

```
log4j.appender.RollingFile.File=${INCX_HOME}/log/logmanager.log
```

Event Logger – The event logger logs events and commands run on a Sapphire appliance to an Executive event log. The event logger can be configured using the eventlogger_log4j.properties file in the $INCHOME directory. The default logging output location is $INCHOME/log/eventlogger.log and the default categories and corresponding severities are:

```
log4j.category.com.diamondip.management=DEBUG
log4j.category.com.diamondip.ipcontrol=INFO
```

Management Logger – The management logger logs commands and status checks from the Executive to deployed Sapphire appliances. This log can be configured using the mgmt_log4j.properties file in the $INCHOME directory. The default logging output location is $INCHOME/log/mgmt.log and the default categories and corresponding severities are:

```
log4j.category.com.diamondip.management=DEBUG
log4j.category.com.diamondip.common=DEBUG
```

Agent and Messaging Services (Executive and Remote)

An agent runs on both the Executive as well as on each remote server. They communicate with each other using IPAM messaging services.

Agents – Configure agent logging in the $INCHOME/agent_log4j.properties file, which defaults to $INCHOME/log/agent.log. The default logging output destination is specified on the line:

```
log4j.appender.RollingFile.File=${INCX_HOME}/log/agent.log
```

The categories of events and corresponding severities can be specified in this file:

```
log4j.category.com.diamondip.netcontrol=DEBUG
```

```
log4j.category.com.diamondip.ipcontrol=DEBUG
log4j.category.com.diamondip.netcontrol.task=DEBUG
log4j.category.com.diamondip.netcontrol.agent=DEBUG
log4j.category.com.diamondip.common.util.Shell=DEBUG
```

Messaging – The messaging service of IPAM is provided by the ActiveMQ service. Logging for ActiveMQ is configured by the `$INCHOME/activemq/conf/log4j.properties` file. The default output destination is $INCHOME/log/activmq.log as specified in the line:

```
log4j.appender.out.file=${INC_HOME}/log/activemq.log
```

DHCP/DNS Services (Remote)

DHCP

The ISC DHCP service logs by default to `/var/log/daemon.log`. DHCP log messages may be directed to a specified syslog facility using the `log-facility` *`facility`*`;` command. The *`facility`* argument, daemon by default as mentioned, may be defined as any of the following: auth, authpriv, cron, daemon, ftp, kern, local0 through local7, lpr, mail, mark, news, ntp, security, syslog, user, and uucp. In addition to setting this value, you may need to modify your syslog.conf file to configure logging of the DHCP server. For example, you might add a line like this:

```
local7.debug /var/log/dhcpd.log
```

The syntax of the syslog.conf file may be different on some operating systems - consult the syslog.conf manual page on your system to be sure. To get syslog to start logging to the new file, you must first create the file with correct ownership and permissions (usually, the same owner and permissions of your /var/log/messages or /usr/adm/messages file should be fine) and send a SIGHUP to syslogd. Some systems support log rollover using a shell script or program called newsyslog or logrotate, and you may be able to con- figure this as well so that your log file doesn't grow uncontrollably.

Because the log-facility setting is controlled by the dhcpd.conf file, log messages printed while parsing the dhcpd.conf file or before parsing it are logged to the default log facility. To pre- vent this, see the README file included with this distribution, which describes how to change the default log facility. When this parameter is used, the DHCP server prints its startup message a second time after parsing the configuration file, so that the log will be as complete as possible.

DNS

BIND defines logging categories which provide classification of loggable events for the server. This provides a convenient way to direct logging events for different categories of events to different destinations. For example, you may want to log critical events logged for the security category to the syslog daemon, while warn events for query category are logged to a log file. The logging destinations and associated output formatting for each category are defined as logging channels within named.conf.

Each logging channel is defined by the specifying:

- output destination –

  `file` – an appended log file on the server

  `syslog` - a protocol for sending log messages over an IP network

  `stderr` - standard error – the operating system standard output stream for error messages

  `null` - discard

- severity – `critical`, `error`, `warning`, `notice`, `info`, `debug` [*`level`*], or `dynamic` - ordered highest to lowest; listed severity indicates that severity level and higher
- additional information to include with the event output (default is to not include any of the following):
  - include the logging category in the output or not (`print-category`)
  - include the severity in the output or not (`print-severity`)
  - include the timestamp in the output or not (`print-time`)
- For output destinations of `stderr` or `null`, no further parameters are needed; however for `file` or `syslog` output, additional parameters may be defined:
- for `file`, specify file parameters:
  - `path_name` of the file
  - the number of rollover `versions` to keep – as a file maxes out in size, a copy is created with an integer suffix of the version number, i.e., 1, 2, etc.
  - maximum `size` of each file, specified in bytes, with k, m, or g prefix for KB, MB, and GB respectively
- for `syslog`, specify the syslog facility (the type of program that is logging the message - make sure the server OS supports the selected facility)
  - `kern` – kernel messages
  - `user` – user-level messages
  - `mail` – mail subsystem
  - `daemon` – system daemon
  - `auth` – security/authorization messages (deprecated in recent syslogd versions)
  - `syslog` – syslogd messages
  - `lpr` – line printer subsystem
  - `news` – USENET news subsystem
  - `uucp` – UUCP subsystem
  - `cron` – timed cron daemon
  - `authpriv` – security/authorization messages - private
  - `ftp` – file transfer protocol
  - `local0`, `local1`, `local2`, `local3`, `local4`, `local5`, `local6`, `local7` – local facilities

The syntax for defining logging on a BIND server falls within the logging statement block and defines each channel, then for each BIND-defined category, associates one or more channels over which to direct corresponding category events of the defined severity.

```
logging {
   [ channel channel_name {
      ( file path_name [versions ( number | unlimited )] [ size size] |
        syslog facility |
        stderr |
        null );
      [ severity severity; ]
      [ print-category (yes | no); ]
      [ print-severity (yes | no); ]
      [ print-time (yes | no); ]
   }; ]
   [category category_name {
      channel_name; [ channel_name; . . . ]
   }; ]
   . . .
};
```

where :

- *channel_name* = a user defined name for the logging channel
- *path_name* = file path
- *number* = integer number of file versions
- *size* = max log file size, e.g., 100k, 20m, 5g, etc.
- *facility* = (kern | user | mail | daemon | auth | syslog | lpr | news | uucp | cron | authpriv | ftp | local0 | … local7)
- *severity* = (critical | error | warning | notice | info | debug [level] | dynamic)
- *category_name* = one of the pre-defined BIND categories. Valid values include:
  - default – defines the channel parameters for those categories for which no explicit channel parameters have been defined.
  - general – a category for those events that don't fall into other defined categories
  - database – events related to server databases for storing zone and cache data
  - security – events related to approval and denial of requests
  - config – configuration file processing events
  - resolver – events related to resolution activities on behalf of resolvers, e.g., iterative lookups performed
  - xfer-in – incoming zone transfer events
  - xfer-out – outgoing zone transfer events
  - notify – events related to notify messages
  - client – client/resolver events

- `unmatched` – events triggered by the inability of the server to identify the view or corresponding class for which a given query applies
- `network` – network related events
- `update` – events related to dynamic update transactions
- `update-security` – dynamic update request events, e.g., approvals and denials
- `queries` – queries received by the server, including the querierer's IP address and port number, as well as query owner name, class and type, along with header information including the RD (recursion desired) flag setting, use of EDNS, and if the query was signed.
- `dispatch` – server module hand-off events
- `dnssec` – events related to DNSSEC and TSIG processing
- `lame-servers` – events identifying a lame delegation where a delegated server is unable to resolve or further process a query
- `delegation-only` – queries forced to NXDOMAIN due to a delegation-only zone or a delegation-only in a hint or stub zone declaration.
- `edns-enabled` – queries that have been sent to other servers using EDNS but were resent without using EDNS due to timeouts awaiting a response from the EDNS query. This may indicate other DNS servers which are not responding due to packet loss or neglect in replying with an error code.

For more information on CPNR DHCP/DNS services logging, refer to the "Logging Server Events" section of CPNR User Guide or CPNR Administration Guide.