Perform Routine Maintenance

- Cisco Finesse Services, on page 1
- Log Collection, on page 2
- Collect Logs using Cisco Unified Real-Time Monitoring Tool, on page 5
- Finesse Performance Counters, on page 9
- JMX Counter Thresholds, on page 10
- Remote Account Management, on page 12

Cisco Finesse Services

You can access the following Finesse services from the CLI:

- **Cisco Finesse Notification Service:** This service is used for messaging and events. If this service is not started, you cannot view call events, agent state changes, or statistics, and the Finesse Desktop will not load after sign-in.
- **Cisco Finesse Tomcat:** This service contains all deployed Finesse applications. A restart of the Cisco Finesse Tomcat service requires that all agents sign out and sign back in.

The deployed applications in the Cisco Finesse Tomcat service include:

- **Finesse Desktop application:** Provides the user interface for agents and supervisors.
- **Finesse Rest API application:** Provides integration with the Cisco CTI Server for the Finesse desktop and Finesse administration application. The APIs available to a user depends on the role associated with that user's credentials. This application also provides a programming interface that can be used by third-party applications that are written to use the Finesse REST API.
- **Finesse Administration application:** Provides the administrative operations for Finesse.

If a Cisco Finesse service-related problem exists, restart a Finesse service as a last resort. Most service-related problems cannot be corrected by restarting a service. Restart A Cisco DB only if the service is down.

Note To restart the Cisco Finesse Notification Service, you must stop and start services in the following order:

1. Stop the Cisco Finesse Tomcat service.
2. Stop the Cisco Finesse Notification Service.
3. Start the Cisco Finesse Notification Service.
4. Start the Cisco Finesse Tomcat service.

View, Start, or Stop Services

Procedure

Step 1 Sign in to the CLI using the credentials for the Administrator User account.

Step 2 To view a list of all services and their states, enter the following command: **utils service list**.

Services are shown in one of the following states: STOPPED, STARTING, or STARTED.

STOPPED means the service is not running. STARTING means the service is starting operation and performing any initialization. STARTED means the service has successfully initialized and is operational.

Step 3 To start a service, enter the following command: **utils service start** *service name*.

Example:

For example, to start Cisco Finesse Tomcat, enter the command **utils service start Cisco Finesse Tomcat**.

Step 4 To stop a service, enter the following command: **utils service stop** *service name*.

Example:

For example, to stop Cisco Finesse Tomcat, enter the command **utils service stop Cisco Finesse Tomcat**.

Log Collection

These commands prompt you to specify a secure FTP (SFTP) server location to which the files will be uploaded.

To obtain logs:

- Install log: **file get install desktop-install.log**

  Use this command to see the installation log after the system is installed.

  This log is written to the SFTP server and stored as a text file written to this path: *<IP Address>\<date time stamp>\install\desktop-install.log*

- Desktop logs: **file get activelog desktop recurs compress**

Use this command to obtain logs for the Finesse web applications. This command uploads a zip file that contains the following directories:

- **webservices:** Contains the logs for the Finesse backend that serves the Finesse REST APIs. The maximum size of an uncompressed desktop log file is 100 MB. The maximum size of this directory is approximately 4.5 GB. After a log file reaches 100 MB, that file is compressed and a new log file is generated. Output to the last compressed desktop log file wraps to the log file created next. The log file wrap-up duration can vary, based on the number of users on the system. Timestamps are placed in the file name of each desktop log.
- **desktop:** Contains logs from the Finesse agent desktop gadget container that holds the Finesse desktop gadgets. Any container-level errors with Finesse agent desktop will appear in these log files.
- **admin:** Contains logs from the Finesse administration gadget container that holds the administration gadgets. Any container-level errors with the Finesse administration console appear in these log files.
    - **audit-log:** Audit logs contain all admin operations (including Finesse admin UI and REST client operations) and supervisor operations for Team Message. The maximum size of an uncompressed audit log file is 100 MB. The maximum size of total audit log files (including compressed log files) is approximately 1 GB. After a log file reaches 100 MB, that file is compressed and a new log file is generated. The log file wrap-up duration can vary, based on the number of users on the system. The log contains the following parameters:
        - Timestamp
        - User Id of the administrator
        - Method of operation (PUT, POST, DELETE). GET operations will not be logged
        - URL
        - Payload
- **clientlogs:** Contains the client-side logs that are submitted from the Cisco Finesse agent desktop to the Finesse server. Each log file is no larger than 1.5 MB and contains a timestamp and the agent ID of the agent who submitted the file. A new log file is created each time that an agent submits client-side logs (the data is not appended to an existing log file). The maximum size of this directory is 100 MB. The directory holds a maximum number of 25000 clientlog files. When the directory exceeds the size limit or the file count, the oldest files are deleted.
- **openfireservice:** Contains startup and shutdown-related information logs for the Cisco Finesse Notification Service.
- **openfire:** Contains limited error and information logs for the Cisco Finesse Notification Service.
- **realm:** Contains the logs for authentication requests from clients that are handled by the Finesse backend.
- **db:** Contains the Finesse database logs.
- **/finesse/logs:** Contains the logs for the Cisco Finesse Tomcat service.
- **jmx:** Contains the JMX counters data that is generated by the JMX logger process. It contains important jmx counters that are exposed by Finesse and openfire.

- **finesse_maintenance_mode.log:** Contains the logs of Cisco Finesse hook script implementation of orchestration manager.

These logs are stored in the following path on the SFTP server: *<IP address>\<date time stamp>\active_nnn.tgz* , where `nnn` is timestamp in long format.

- WebProxy Service logs: **file get activelog webproxy recurs compress**

  Use this command to obtain logs for the WebProxy Service. The maximum size of an uncompressed webproxy log file is 10 MB. The maximum size of this directory is approximately 500 MB. After a log file reaches 10 MB, that file is compressed and wraps to the new log file which is generated. The log file wrap-up duration can vary, based on the number of users on the system. Timestamps are placed in the file name of each webproxy log.

  These logs are stored in the following path on the SFTP server: *<IP address>\<date time stamp>\active_nnn.tgz* , where `nnn` is timestamp in long format.

  This command uploads a zip file that contains the following log files:

  - **access.log**: Contains the webproxy access logs after you configure the access log-level using the **set webproxy access-log-level** CLI. For more information on CLI commands, see *WebProxy Service*.
  - **error.log**: Contains the webproxy error logs.
  - **webproxy_cli.log**: Contains the webproxy CLI logs. For more information on CLI commands, see *WebProxy Service*.
  - **webproxy_launcher.log**: Contains the logs after the WebProxy Service is launched.

Note To access the individual log file, use the command **file get activelog webproxy/<log filename>**.

For example, **file get activelog webproxy/error.log**

- Servm log: **file get activelog platform/log/servm*.* compress**

  Use this command to obtain logs that are generated by the platform service manager that manages the starting and stopping of the Finesse services.

  The desktop and servm logs are compressed to one set of files.

  These logs are stored to the following path on the SFTP server: *<IP address>\<date time stamp>\active_nnn.tgz* , where `nnn` is the timestamp in long format.

- Platform Tomcat logs: **file get activelog tomcat/logs recurs compress**

  These logs are stored to the following path on the SFTP server: *<IP address>\<date time stamp>\active_nnn.tgz* , where `nnn` is the timestamp in long format.

- Install log: **file get install install.log**

  These logs are stored to the following path on the SFTP server: *<IP address>\<date time stamp>\active_nnn.tgz* , where `nnn` is timestamp in long format.

Note Log collection may fail when you use the compress flag if there are a lot of log files. If collection fails, run the command again without the compress flag.

Call Variables Logging

From Cisco Finesse Release 12.5(1) onwards, the call variables logging in Cisco Finesse logs are disabled by default. The callVariables contain sensitive user information and this property allows the administrator to decide whether the information must be captured in the logs. You can enable the call variables logging by using the CLI commands.

Collect Logs using Cisco Unified Real-Time Monitoring Tool

Cisco Finesse supports the Cisco Unified Real-Time Monitoring Tool (RTMT) for log collection. Use the following procedure to collect logs using Unified RTMT.

Note Finesse supports RTMT only for log collection. Other RTMT features are not supported.

Before you begin

Download and install RTMT on a client computer from the following URL:

https://*FQDN*:8443/plugins/CiscoRTMTPlugin.zip

where *FQDN* is the Fully Qualified Domain Name of the Finesse server.

Procedure

Step 1 Log in to Unified RTMT using Finesse administrator credentials.

Step 2 In the tree hierarchy, select **Trace & Log Central.**

Step 3 Double-click **Collect Files**.

The Trace Collection wizard appears.

Step 4 Select the services and Finesse nodes for which you want to collect logs, and complete the wizard.

What to do next

For detailed instructions, see the *Cisco Unified Real-Time Monitoring Tool Administration Guide*, which is listed here:

https://www.cisco.com/c/en/us/support/unified-communications/unified-communications-manager-callmanager/products-maintenance-guides-list.html

Syslog Support for Critical Log Messages

Cisco Finesse generates syslogs for critical log messages. Use the following procedure to view the logs using Unified RTMT.

Before you begin

Download and install RTMT on a client computer from the following URL: `https://FQDN:8443/plugins/CcmServRtmtPlugin.exe`, where FQDN is the Fully Qualified Domain Name of the Finesse server.

Procedure

Step 1 Log in to Unified RTMT using Finesse administrator credentials.

Step 2 In the tree hierarchy, select **SysLog Viewer** or choose **System** > **Tools** > **SysLog Viewer** > **Open SysLog Viewer**.

Step 3 From the **Select a Node** drop-down list, choose the server where the logs that you want to view are stored.

Step 4 Under the **Logs** tab, select **Application Logs** > **CiscoSyslog** to view and save the syslog file.

Tip
When you double-click the CiscoSyslog message, the **Show Detail** dialog displays the syslog definition and recommended actions in an adjacent pane.

For more information, see the *Cisco Unified Real-Time Monitoring Tool Administration Guide*.

Note
System log messages generated by Cisco Finesse are also available under **SysLog Viewer** > **System Logs** > **messages**.

The following are the different types of messages and corresponding descriptions that are captured in the **SysLog Viewer** > **System Logs** > **messages**.

- CTI_SOCKET_ERROR

  System has encountered an error connecting to the CTI server.

- CTI_CONNECTION_LOST

  System has lost contact with the CTI server.

- CTI_OPEN_FAILURE

  CTI Server rejected open request.

- CTI_CONNECTION_RETRIES_EXCEEDED

  System has failed to connect to the CTI server in the allowed number of retries.

- CTI_CONNECTION_ESTABLISHED

  System has successfully connected to the CTI server.

- SUBSYS_INIT_ERROR

  Error initializing subsystem.

- UNABLE_TO_CONNECT_TO_XMPP_SERVER

Unable to connect xmpp server.

- DB_SS_CONNECTION_CHECK

  There was an error connecting to the database.

- cfservice_CORE_ERROR_DB_CONNECTION

  Unable to connect to the Database.

- AWDB_NOT_ACCESSIBLE

  Unable to connect to AWDB server.

- VOS_DB_ADAPTER_ERROR

  There was an error on the VOS DB Adapter operation.

- FINESSE_APP_STARTUP_ERROR

  Error during Finesse Application Startup.

- OF_STATE_CHANGED

  OF subsystem state successfully changed.

- CONNECTED_TO_XMPP_SERVER

  Successfully connected to xmpp server.

- SSO_API_ERROR

  Error processing REST API Request for SSO.

- API_ERROR_DETAIL

  Error processing REST API request.

- AWDB_CONNECTION_ERROR

  Error while connecting the AWDB server.

- AWDB_CONNECTION_SWITCH_SUCCESS

  AWDB server connection successfully switched.

- CTI_DISCONNECT

  Disconnecting primary CTI connection.

- STANDBY_CTI_DISCONNECT

  Disconnecting standby CTI connection.

- RESET_STANDBY_CONNECTION

  Resetting standby CTI connection.

- RESET_PRIMARY_CTI_CONNECTION

  Resetting primary CTI connection.

- SWAP_CTI_CONNECTIONS

  Swapping primary and standby CTI Connections.

- STANDBY_ACMI_CONNECTION_ESTABLISHED

  Established connection with standby CTI.
- STANDBY_ACTIVE_EVENT

  Received standby active event from standby CTI.
- MAINTENANCE_MODE_REQUEST_TIMEOUT

  CTI STANDBY_ACTIVE_EVENT isn't received before the threshold time.
- MAINTANCE_MODE_CONTINUE_CTI

  CTI confirms to continue maintenance mode.
- MAINTANCE_MODE_DUPLICATE_REQUEST

  Duplicate maintenance mode request received from CTI.
- FINESSE_MAINTENANCE_MODE_FAILED

  Finesse current node went out of service.
- FINESSE_MAINTENANCE_MODE_FAILED

  Finesse alternate node is in maintenance mode.
- FINESSE_MAINTENANCE_MODE_FAILED

  Internal error occurred while performing the maintenance mode operation.
- FINESSE_MAINTENANCE_MODE_FAILED

  Finesse couldn't migrate these agents {count}
- FINESSE_MAINTENANCE_MODE_FAILED

  Finesse Maintenance Mode operation failed with reason {reason}
- FINESSE_MAINTENANCE_MODE_REQ_REJECTED Finesse OUT_OF_SERVICE

  Rejecting the maintenance mode request.
- FINESSE_MAINTENANCE_MODE_REQ_REJECTED

  CTI maintenance is in progress.
- FINESSE_MAINTENANCE_MODE_REQ_REJECTED

  Finesse maintenance is already completed.
- FINESSE_MAINTENANCE_MODE_REQ_REJECTED

  This is a single node setup.
- FINESSE_MAINTENANCE_MODE_REQ_REJECTED

  The alternate node of Finesse is OUT_OF_SERVICE.
- FINESSE_MAINTENANCE_MODE_REQ_REJECTED

  The alternate node of Finesse is in maintenance mode.
- FINESSE_MAINTENANCE_MODE_REQ_ACCEPTED

Changing Finesse Maintenance Mode to IN_PROGRESS.

- FINESSE_MAINTENANCE_MODE_COMPLETED

  Finesse successfully migrated all the agents to the alternate node.

- DRAPI_REST_OPERATION_JOB_DISCARD

  Clearing the Digital Routing API rest job when operation has failed multiple times and the maximum number of retries are over.

- DRAPI_HOST_ALERT

  Failover of Digital Routing API host-pair.

  Failover isn't supported when the Digital Routing API host backup isn't configured.

- DRAPIAsyncRestClient

  Failed to create SSL connection to Digital Routing API.

- THIRD_PARTY_GADGET_MONITORING_SERVICE_ERROR

  Error starting 3rdPartyGadget monitoring service = {Error Name}

Note
By default, Finesse tomcat supports listening upto 8000 folders including subdirectories in the `finesse/3rdpartygadget` folder. If you add folders beyond 8000, it will not be reflected nor monitored and the syslog errors are as follows:

- While starting Finesse tomcat, if the number of folders and subfolders are more than 8000, the error `Error while starting a third-party gadget monitoring service` is displayed.
- While Finesse tomcat running, when new folders/subfolders are added in the `3rdpartygadget` folder, if the count of folders is is more than 8000, the error `Error occurred in a third-party gadget monitoring service` is displayed.

Finesse Performance Counters

You can get the JMX counters from Finesse using the following methods of Performance API:

- list
- read

For more information about these methods, refer to the *Cisco FinesseWeb Services Developer and JavaScript Guide* on the DevNet site.

For the list of JMX counters that are available for digital channels, see the *JMX Counters for Digital Channels* table in the *JMX Counter Thresholds* topic.

JMX Counter Thresholds

Cisco Finesse provides important JMX counters with associated threshold values that can be used to monitor the health of Finesse. The following tables list the JMX counters with corresponding threshold values at the login phase and steady phase (the time after the agents have logged in).

Table 1: JMX Counters on Tomcat Processes (Port 12399) - Login Phase Thresholds

JMX Counter	Description	JMX Object Nam
ThreadCount	The number of threads running at the current moment.	java.lang:type =
PeakThreadCount	The maximum number of threads run at the same time since the JVM was started or the peak was reset.	java.lang:type =
currentThreadCount	The number of threads the thread pool currently has (both busy and free).	Catalina:type = T
currentThreadsBusy	The number of threads currently processing requests.	Catalina:type = T
RequestLongestTime	The maximum amount of time taken to complete an API request, in milliseconds.	com.cisco.ccbu:c
processCPULoad	The CPU load in this process.	java.lang:type =
NumOfActiveAgentsLoggedIn	The number of agents logged in with XMPP Presence as available in the current side.	com.cisco.ccbu:c
NumOfAgentsLoggedIn	The number of agents and supervisors logged in currently.	com.cisco.ccbu:c

Table 2: JMX Counters for Digital Channels

Counter Name	Description	JMX Object Name
CloseTasksRequestsInterval	Close task API requests during interval. **Note** This does not include retry request.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats
TotalCloseTasksRequests	Total number of Close task API requests to Digital Routing API from starting of the Finesse service. **Note** This does not include retry request.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats

Counter Name	Description	JMX Object Name
TransferTasksRequestsInterval	Transfer task API requests during interval. **Note** This does not include retry request.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats
TotalTransferTasksRequests	Total number of Transfer task API requests to Digital Routing API from starting of the Finesse service. **Note** This does not include retry request.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats
DiscardedTasksInterval	Discarded Digital Routing API tasks during interval.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats
TotalDiscardedTasks	Total number of discarded Digital Routing API tasks from starting of the Finesse service.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats
FailoversInInterval	Failovers during intervals.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats
TotalFailovers	Total number of Digital Routing API failovers from starting of the Finesse service.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats
PendingRequestsInQueue	Total number of Digital Routing API rest requests that are pending in queue.	com.cisco.ccbu:category= DRAPITasksStatsBean,component0= DRAPITasksStats

Table 3: JMX Counters on Tomcat Processes (Port 12399) - Steady Phase Thresholds

JMX Counter	Description	JMX Object N
ThreadCount	The number of threads running at the current moment.	java.lang:typ
PeakThreadCount	The maximum number of threads run at the same time since the JVM was started or the peak was reset.	java.lang:typ
TotalCallsInSystem	The total number of active calls in the system.	com.cisco.cc

JMX Counter	Description	JMX Object Nam
AverageProcessingTime	The average time taken for processing CTI messages, in milliseconds.	com.cisco.ccbu:c
currentThreadCount	The number of threads the thread pool currently has (both busy and free).	Catalina:type = T
currentThreadsBusy	The number of threads currently processing requests.	Catalina:type = T
RunnablesQueued	Runnables (CTI Messages) still queued.	com.cisco.ccbu:c
TasksQueued	The tasks (such as client requests and CTI messages) queued.	com.cisco.ccbu:c
RequestLongestTime	The maximum amount of time taken to complete an API request, in milliseconds.	com.cisco.ccbu:c
processCPULoad	The CPU load in this process.	java.lang:type =

The following table lists the thresholds for counters related to Openfire processes.

Table 4: Counters Related to Openfire (JMX Port 12348)

JMX Counter	Description	JMX Object Nam
ExecutingTaskCount	The number of tasks (messages published to node) that are run currently.	com.cisco.ccbu.f
QueuedTaskCount	The number of tasks in the queue. Messages that are getting published to a node are placed in the queue.	com.cisco.ccbu.f
PeakThreadCount	The maximum number of threads run at the same time since the JVM was started or the peak was reset.	java.lang:type =
ThreadCount	The number of threads running at the current moment.	java.lang:type =
processCPULoad	The recent CPU usage for the Java Virtual Machine process.	java.lang:type =

Remote Account Management

Run the following command to enable, disable, create, and check the status of a remote access account:

utils remote_account

A remote account generates a passphrase that allows Cisco support personnel to get access to the system for the specified life of the account.

- **utils remote_account create** *account life*

  account is the account name. *life* indicates the life of the account in days.
- **utils remote_account disable**
- **utils remote_account enable**
- **utils remote_account status**