Cisco Finesse CLI

- Cisco Finesse Services, on page 1
- Finesse Log Configuration, on page 2
- Toaster Notifications, on page 6
- Finesse IPPA Inactivity Timeout, on page 6
- Configuring Queue Statistics, on page 7
- Cross-Origin Resource Sharing (CORS) , on page 7
- Gadget Source Allowed List, on page 10
- Supported Content Security Policy Directives, on page 12
- Finesse System Commands , on page 13
- Desktop Properties, on page 14
- Service Properties, on page 26
- Log Collection Schedule, on page 31
- Upgrade, on page 32
- Shutdown, on page 34
- Replication Status, on page 34
- View Property , on page 35
- Update Property , on page 35
- Signout from Media Channels, on page 35
- Finesse Maintenance Mode Services, on page 36
- ConnectedUsersInfo, on page 36
- set webapp session maxlimit, on page 37
- set webapp session timeout, on page 38
- Update Cloud Connect Connection Time, on page 39
- AI Services Configuration, on page 40
- Certificate Configuration, on page 41

Cisco Finesse Services

To view, start, or stop services:

- **show network all detail** : View the platform TCP/IP services, UDP services, and Unix domain sockets used by Cisco Finesse:
- **utils service list**: This command retrieves a list of all services and their status.

Services are shown in one of the following states:

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

- **utils service start** *service name*: This command starts the named service.
- **utils service stop** *service name*: This command stops the named service.
- **utils service start Cisco Finesse Tomcat**: This commans starts Cisco Finesse Tomcat.
- **utils service stop Cisco Finesse Tomcat**: This commans stops Cisco Finesse Tomcat.
- **utils service restart Cisco Finesse Tomcat**: This commans restarts Cisco Finesse Tomcat.

Note If a Cisco Finesse service-related problem exists, restart the Finesse service. Note that most service-related problems cannot be corrected by restarting a service.

Finesse Log Configuration

Use the following CLI commands to add, delete, update, or view the logger configuration in the system for Finesse.

utils finesse log configuration add

Creates a custom log configuration in the Finesse system. The logs record information about the encountered issues of different severity levels for a specific Finesse module.

Command Syntax

utils finesse log configuration add *[module] [name] [level]*

Options

Note
- Adding multiple module names, log configuration names, and log configuration level values are not supported.
- Log configuration with name ROOT is not allowed.

- *module*—Unique name of Finesse module for which log configuration has to be added. The module name is case sensitive. The following are the valid Finesse modules.
  - *admin*—Finesse administration module.
  - *audit*—Finesse audit module for all administration (including Finesse admin UI and REST client) and supervisor operations.
  - *desktop*—Finesse desktop module.

- *diagnostics*—Finesse diagnostics module.
- *FIPPA*—Finesse IP Phone Agent (IPPA) application module.
- *realm*—Finesse realm module.
- *shindig*—Shindig web application module.
- *valve*—Finesse valve module.
- *webservices*—Finesse webservices module.

- *name*—Package name or fully qualified class name of the Finesse application. The name is case sensitive.
- *level*—Defines the different severity level associated with the log configuration. The following are the valid log configuration levels.
  - *OFF*—Turns off the severity level.
  - *ERROR*—Sets the severity level to error.
  - *WARN*—Sets the severity level to warning.
  - *INFO*—Sets the severity level to information.
  - *DEBUG*—Sets the severity level to debug.
  - *TRACE*—Sets the severity level to trace.
  - *ALL*—Sets the severity level to all.

Note Setting the log configuration level to DEBUG or TRACE impacts system performance. This must be done in consultation with Cisco support to ensure that the modules with high log output are not be enabled with TRACE levels in production severs.

Example

The following is the sample output for creating the log configuration named *com.cisco.cc.common.subsystem* under the Finesse *webservices* module with log configuration level as *DEBUG*.

```
admin:utils finesse log configuration add webservices com.cisco.cc.common.subsystem DEBUG

Warning: Creating the custom log configurations may affect the performance of the Finesse
system.

Press ENTER to continue. Press any other key to exit :

Creating the log configuration, please wait...

Successfully added the log configuration. Changes might take approximately 30 seconds to
take effect..
```

utils finesse log configuration update

Updates an existing custom log configuration in the Finesse system.

Note

- Updating multiple module names, log configuration names, and log configuration level values are not supported.
- Audit log configuration cannot be updated.

Command Syntax

utils finesse log configuration update *[module] [name] [level]*

Options

- *module*—Unique name of Finesse module for which log configuration has to be updated. The module name is case sensitive. For more information on the Finesse modules, see utils finesse log configuration add.
- *name*—Package name or fully qualified class name of the Finesse application. The name is case sensitive.
- *level*—Defines the different severity level associated with the log configuration. For more information on the severity levels, see utils finesse log configuration add.

Note Setting the log configuration level to DEBUG or TRACE impacts system performance.

Example

The following is the sample output for updating the log configuration named *com.cisco.cc.common.subsystem* under the Finesse *webservices* module with log configuration level as *TRACE*.

```
admin:utils finesse log configuration update webservices com.cisco.cc.common.subsystem TRACE

Warning: Updating the log configuration level to DEBUG or TRACE may affect the performance
 of the Finesse system.

Press ENTER to continue. Press any other key to exit :

Updating the log configuration, please wait...

Successfully updated the log configuration. Changes might take approximately 30 seconds to
 take effect.
```

utils finesse log configuration delete

Deletes an existing custom log configuration in the Finesse system.

Note

- ROOT log configurations cannot be deleted.
- Deleting multiple log configuration names are not supported.

Command Syntax

utils finesse log configuration delete *[module] [name]*

Options

- *module*—Unique name of the Finesse module. The module name is case sensitive.
- *name*—Package name or fully qualified class name of the Finesse application. The name is case sensitive.

Example

The following is the sample output for deleting the log configuration named *com.cisco.cc.common.subsystem* under the Finesse *webservices* module.

```
admin:utils finesse log configuration delete webservices com.cisco.cc.common.subsystem
Deleting log configuration, please wait...

Successfully deleted the log configuration. Changes might take approximately 30 seconds to
 take effect.
```

utils finesse log configuration list

Lists all log configurations in the Finesse system.

Command Syntax

utils finesse log configuration list

Example

The following is the sample output for all the log configuration in the Finesse system.

```
admin:utils finesse log configuration list
Requesting log configurations, please wait...
Below is the list of log configurations in Finesse.
+---+-----------+-------------------------------------------------------------------+-------+
| No. |     Module     |                                                                Name
                                    |  Level  |
+---+-----------+-------------------------------------------------------------------+-------+
| 1.  | admin          | ROOT
                                    | INFO    |
| 2.  | audit          | ROOT
                                    | INFO    |
| 3.  | desktop        | ROOT
                                    | INFO    |
| 4.  | diagnostics    | ROOT
                                    | INFO    |
| 5.  | FIPPA          | ROOT
                                    | INFO    |
| 6.  | realm          | ROOT
                                    | INFO    |
| 7.  | shindig        | ROOT
                                    | INFO    |
| 8.  | valve          | ROOT
                                    | INFO    |
| 9.  | webservices    | ROOT
                                    | INFO    |
| 10. | FIPPA          | org.jivesoftware
                                    | WARN    |
| 11. | webservices    | org.hibernate
                                    | INFO    |
| 12. | webservices    | com.cisco.cc.common.subsystem
                                    | TRACE   |
+---+-----------+-------------------------------------------------------------------+-------+
```

Toaster Notifications

Toaster notifications are enabled by default after a fresh installation of Cisco Finesse. Use the following CLI commands to disable, enable, and check the status of the toaster notifications:

- **utils finesse toaster enable [closeTimeout]**: This command enables the Cisco Finesse toaster notification.

  While enabling toaster notification, use the **closeTimeout** parameter (timeout in seconds) to set the time interval after which toaster automatically closes. If no parameter is specified, timeout is set to 8 seconds by default. The valid range for timeout activity is between 5-15 seconds. The browser must be refreshed for timeout changes to take effect.

Note The configured timeout for browser notifications depends on the operating system and browser settings. The timeout value is honored in Chrome browser in Windows OS. However, the other supported browsers do not honor the configured notification timeout value consistently.

- **utils finesse toaster disable**: This command disables the Cisco Finesse toaster notification.
- **utils finesse toaster status**: This command displays the status (enable or disable) of the Cisco Finesse toaster notification.

Note Cisco Finesse Toaster Notification does not work with Internet Explorer browser.

Finesse IPPA Inactivity Timeout

Use the following CLI commands to enable or disable the Inactivity Timeout feature in Finesse IPPA. You must either disable the Finesse Inactivity Timeout feature or increase the timeout in the range of 120 seconds to one day (in seconds), so that the Finesse IPPA agent is not logged out if on any other screen:

- **utils finesse ippa_inactivity_timeout enable**: This command enables Finesse IPPA Inactivity Timeout.

Note The default time set for inactivity timeout is 120 seconds.

- **utils finesse ippa_inactivity_timeout disable**: This command disables Finesse IPPA Inactivity Timeout.

Note When inactivity timeout is disabled, you will not be logged out of Finesse IPPA, if the agent is on any other screen.

- **utils finesse ippa_inactivity_timeout enable inactivity_timeout**: This command enables the Finesse IPPA Inactivity Timeout with timeout set to n seconds.

Note Minimum value of n must be 120 seconds and maximum value can be up to one day (86400 seconds).

- **utils finesse ippa_inactivity_timeout status**: This command checks the status of Finesse IPPA Inactivity Timeout.

Note The Finesse IPPA Inactivity Timeout CLIs should be run on primary and secondary Finesse servers. Enabling or disabling this feature requires a restart of Cisco Finesse Tomcat, and restart must be done in the maintenance window. During upgrade, the inactivity timeout configuration is not retained and should be re-configured post upgrade.

To know how this feature works on specific IP phone models, see https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-device-support-tables-list.html

Configuring Queue Statistics

The Queue Statistics gadget is enabled by default as part of Cisco Finesse new installation (Unified CCE only). When performing a system upgrade from Cisco Finesse 11.5(1), the desktop custom layout needs to be modified by the administrator for the Queue Statistics gadget to be displayed on the Agent and Supervisor desktop.

Use the following CLI commands to enable and disable the queue statistics polling or check the status of the queue statistics polling:

- **utils finesse queue_statistics enable**
- **utils finesse queue_statistics disable**
- **utils finesse queue_statistics status**

After performing a system upgrade, during switch-version the queue statistics polling will be enabled by default. The procedure to disable the queue statistics polling remains the same.

Note When enabled, Queue Statistics supports a maximum of 2000 users (Agents and Supervisors).

Cross-Origin Resource Sharing (CORS)

CORS support to the third-party web server is disabled by default for Cisco Finesse and OpenFire. Use the following CLIs to enable CORS for Cisco Finesse and OpenFire and configure the allowed origin list:

Note CORS support to third-party clients is enabled for all origins by default in Cisco Finesse and OpenFire. This corresponds to the **enable_all** mode.

Important After you make changes to the CORS status or to the allowed origin list, restart Cisco Finesse Tomcat and Notification Services for the changes to take effect.

- **utils finesse cors enable**: This command allows CORS for Cisco Finesse APIs and OpenFire requests for allowed origin list. It responds to browser CORS preflight requests and allows valid domains to make Finesse API/OpenFire requests.

  Note
  - Use the **utils finesse cors allowed_origin** CLI to customize the allowed origin list.
  - Any custom headers used in the CORS requests must be added using **utils finesse cors allowed_headers** CLI.

- **utils finesse cors enable_all**: This command allows all origins to make cross domain requests. It responds and allows CORS preflight requests from any domain to make Finesse API/OpenFire requests.

  Note This isn't a secure configuration and is included only to support backward compatibility.

- **utils finesse cors disable**: This command restricts CORS for Cisco Finesse APIs and OpenFire requests. It disallows or prevents CORS preflight requests from any external domain to make Finesse API and OpenFire requests.

  Note If the allowed origin list is already present, the list is preserved and used when CORS is enabled. The CLI changes are reflected only after you clear your cache and close and reopen the browser.

- **utils finesse cors status**: This command displays the CORS status (enable_all, enabled, or disabled) on the console.

For allowing any other header, the following set of CLI commands are added to enable CORS for both Cisco Finesse and OpenFire and to configure the allowed origin list:

- **utils finesse cors allowed_origin list**: This command lists all the origins in the allowed origin list.
- **utils finesse cors allowed_origin add**: This command adds origins to the allowed origin list. Origins can be added by using a comma-separated string. For example:

  utils finesse cors allowed_origin add https://origin1.com:[port]

utils finesse cors allowed_origin add https://origin1.com: [port], https://origin2.com:[port]

Note
- The wildcard character star (*) isn't a valid origin in the allowed origin list.
- The maximum number of characters (cumulative) that are permissible in allowed origin is 4000.

- **utils finesse cors allowed_origin delete**: This command deletes origins from the allowed origin list.

Note Delete lists all the origins in the allowed origin list. The origins can be deleted by selecting the appropriate ones from the list. For example:

utils finesse cors allowed_origin delete

1: http://google.com

2: https://www.cisco.com

3: https://def.com

4: https://abc.com:7777

a: all

q: quit

Select the index of origin(s) to be deleted [1-4 or a,q]

By default the following headers are allowed and exposed:

- **allowed_headers**: Content-Type, X-Requested-With, accept, Origin, Authorization, Access-Control-Request-Method, Access-Control-Request-Headers, requestId, Range.
- **exposed_headers**: Access-Control-Allow-Origin, Access-Control-Allow-Credentials, Access-Control-Allow-Methods, Access-Control-Allow-Headers, Access-Control-Max-Age.

Note These headers can't be modified. Custom headers can be added or removed using the following CLIs:

- **utils finesse cors allowed_headers list**: This command lists all the allowed headers for CORS. The list is used to validate incoming requests to Finesse.
- **utils finesse cors allowed_headers add**: This command adds one or more allowed headers for CORS. Multiple headers can be added as a comma-separated string. For example:
  - utils finesse cors allowed_headers add header1
  - utils finesse cors allowed_headers add header1,header2,header3

Note The wildcard character star (*) isn't supported.

- **utils finesse cors allowed_headers delete**: This command lists the choices for deleting the allowed headers. The choice should be an index as displayed in the list of allowed headers. The list provides the option to delete a single header or all configured custom headers. For example:

  utils finesse cors allowed_headers delete

  1: header1

  2: header2

  a: all

  q: quit

  Select the index of the allowed header to be deleted [1-2 or a,q]: 1

- **utils finesse cors exposed_headers list**: This command lists all the exposed headers for CORS. The list will be used by the browser to validate the accessible headers in the response.
- **utils finesse cors exposed_headers add**: This command adds one or more exposed headers for CORS. Multiple headers can be added by a comma-separated string. For example:

  utils finesse cors exposed_headers add header1

  utils finesse cors exposed_headers add header1,header2,header3

Note The wildcard character star (*) isn't supported

- **utils finesse cors exposed_headers delete**: This command lists the choices for deleting the exposed headers. The choice should be an index as displayed in the list of allowed headers. The list provides option to delete a single header or all configured custom headers. For example:

  utils finesse cors exposed_headers delete

  1: header1

  2: header2

  a: all

  q: quit

  Select the index of the exposed header to be deleted [1-2 or a,q]: 1

All CLIs are node specific and must be run on all nodes in the cluster.

Gadget Source Allowed List

Shindig proxies requests from the Finesse desktop to external servers and this introduces the possibility of server side request forgery (SSRF). To prevent SSRF, you can choose to allow outgoing connections for

specified sources to be used in the gadgets by adding URLs to the allowed list. Note that this functionality is disabled by default for Cisco Finesse.

Use the following CLIs to enable or disable Gadget Source allowed list functionality and to configure source(s) in the allowed list:

- **utils finesse gadget_source_check enable**: This command enables allowed list for Cisco Finesse.
- **utils finesse gadget_source_check disable**: This command disables allowed list for Cisco Finesse.
- **utils finesse gadget_source_check status**: This command prints the allowed list status (enabled or disabled) on Cisco Finesse console.
- **utils finesse gadget_source_check allowed_list list**: This command lists all the source(s) in the allowed list.
- **utils finesse gadget_source_check allowed_list add**: This command adds source(s) to the allowed list. For example,
  - **utils finesse gadget_source_check allowed_list add** https://www.abc.com:8445.
  - **utils finesse gadget_source_check allowed_list add** https://www.abc.com:8445, http://www.abc.com.

Note Wildcard character * is not supported.

The allowed list feature does not perform hostname resolutions. The format of the allowed list entry should match the format in which the gadget requests for a resource.

If **utils finesse gadget_source_check** is enabled, you must add the CUIC URLs to **utils finesse gadget_source_check allowed_list** for the stock gadgets to load. For example,

- utils finesse gadget_source_check enable
- utils finesse gadget_source_check allowed_list add https://<CUIC_Pub_FQDN>
- utils finesse gadget_source_check allowed_list add https://<CUIC_Pub_FQDN>:8444
- utils finesse gadget_source_check allowed_list add https://<CUIC_Sub_FQDN>
- utils finesse gadget_source_check allowed_list add https://<CUIC_Sub_FQDN>:8444

If you do not add the CUIC URLs, Finesse Desktop fails to load and an appropriate error message is displayed.

- **utils finesse gadget_source_check allowed_list delete**: This command deletes source(s) from the allowed list. For example:
  - **utils finesse gadget_source_check allowed_list delete**
    - 1: http://origin1:8080
    - 2: https://origin2:7777
    - a: all
    - q: quit

```
Select the index of origin to be deleted [1-2 or a,q]: 1
```

Note All CLIs are node-specific and must be run on all nodes in the cluster.

After any changes are done to gadget source status or to the allowed list, restart Cisco Finesse Tomcat for changes to take effect.

Supported Content Security Policy Directives

Content Security Policy (CSP) is a standardized set of security directives that can inform the browser of the policies to be used to help mitigate various forms of attacks. CSP frame-ancestor policy defines the allowable locations from where the Finesse desktop can be accessed as an embedded HTML content, which can help prevent click-jacking attacks.

Use the following CLI commands to view, add, or delete the frame-access sources in the response header of Cisco Finesse. This ensures that only the configured sources can embed the Cisco Finesse in an iFrame within their HTML pages.

Note Internet Explorer does not support frame-ancestors, and therefore will not block any websites from loading Cisco Finesse within it.

- **utils finesse frame_access_allowed_list add** *[source1,source2]*—This command adds one or more frame sources, thereby allowing the configured sources to embed the Cisco Finesse in their iFrames. Multiple sources can be provided as a comma-separated list. The source should be of the following format:
  - https://<fqdn>:[port]
  - https://IP:[port]
  - https://<fqdn1>:port, https://<fqdn2>:port

Note
- Wildcard character * is also supported for the FQDN and port entries, which indicates that all the legal FQDN or ports are valid.
- The maximum number of characters (cumulative) that are permissible in allowed list is 2000.

```
admin:utils finesse frame_access_allowed_list add
https://www.abc.com:8445,https://*.abc.com,https://*.abc.com:*,https://10.21.255.25

Source(s) successfully added.
Ensure Source(s) is added to the frame access list in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat and Cisco Finesse Notification Service for the changes to
 take effect:
```

```
utils service restart Cisco Finesse Tomcat
utils service restart Cisco Finesse Notification Service
```

- **utils finesse frame_access_allowed_list delete**—This command displays an indexed list of all the configured frame sources that have been allowed to access Cisco Finesse. Enter the corresponding index number to delete a single source or all the configured sources.

```
admin:utils finesse frame_access_allowed_list delete

1: https://www.abc.com:8445
2: https://*.abc.com
3: https://*.abc.com:*
4: https://10.21.255.25
a: all
q: quit

Select the index of source to be deleted [1-4 or a,q]: 1
Sources deleted successfully.

Restart Cisco Finesse Tomcat and Cisco Finesse Notification Service for the changes to
 take effect:
 utils service restart Cisco Finesse Tomcat
 utils service restart Cisco Finesse Notification Service
```

- **utils finesse frame_access_allowed_list list**—This command lists all the frame sources that are allowed to access Cisco Finesse.

```
admin:utils finesse frame_access_allowed_list list

The following source(s) are configured in the frame access list:
1: https://www.abc.com:8445
2: https://*.abc.com
3: https://*.abc.com:*
4: https://10.21.255.25
```

Finesse System Commands

Configure the following Cisco Finesse system CLIs:

Notifications

Use the following CLI commands to enable or disable the Cisco Finesse notifications. By default, this feature is disabled.

- To enable: **utils finesse notification logging enable**
- To disable: **utils finesse notification logging disable**

Node Statistics

Use the following CLI command to view the run-time statistics for the current node.

- To view: **utils finesse node_statistics list**

```
admin:utils finesse node_statistics list

Warning: Running this command frequently will affect system performance.
Press ENTER to continue. Press any other key to exit :

Wait while the statistics (updated every 5 secs) are being fetched...
```

```
The following are the runtime statistics for the current node.

Active Dialogs Count: 0

Active Tasks Count: 0

Average Configured Media per Agent Count: 0

Average Logged in Media per Agent Count: 0

Average Skill Groups per Agent Count: 0

Max Skill Groups per Agent Count: 0

Total Time for Finesse to Start (in seconds): 32

Logged in Agents on current node: 0

Unique Configured Skill Groups per Agent Count: 0
```

For more information, see *RuntimeConfigInfo API Parameters* section in the *Cisco Finesse Web Services Developer Guide* at https://developer.cisco.com/docs/finesse/.

Locked Out Users

Use the following command to list the locked out users:

- To list: **utils finesse locked_out_users list**

```
admin: utils finesse locked_out_users list

Total locked out users : 2

1) agent1
2) agent3
```

If there are no locked out users:

```
admin: utils finesse locked_out_users list

Total locked out users : 0
```

Desktop Properties

Configure the desktop properties using the following CLIs for the features.

Note Refresh the browser for the changes to take effect.

Active Call Details in the Team Performance Gadget

Use the following CLI commands to enable or disable the active call details:

- To enable: **utils finesse set_property desktop showActiveCallDetails true**

- To disable: **utils finesse set_property desktop showActiveCallDetails false**

View History in the Team Performance Gadget

Use the following CLI commands to enable or disable the agent history:

- To enable: **utils finesse set_property desktop showAgentHistoryGadgets true**
- To disable: **utils finesse set_property desktop showAgentHistoryGadgets false**

Force Wrap-Up Reason

Use the following CLI commands to enable or disable the force wrap-up reason:

Note This is applicable to both voice and non-voice channels.

- To enable: **utils finesse set_property desktop forceWrapUp true**
- To disable: **utils finesse set_property desktop forceWrapUp false**

Show Wrap-Up Timer

Use the following CLI commands to show or hide the timer in wrap-up state:

Note This is applicable to both voice and non-voice channels.

- To hide the timer in wrap-up state: **utils finesse set_property desktop showWrapUpTimer false**
- To display the timer in wrap-up state: **utils finesse set_property desktop showWrapUpTimer true**

By default, the value of this property is set to true.

Wrap-Up Timer Count Down

Use the following CLI commands to set the wrap-up timer to count down or count up the time:

Note This is applicable to both voice and non-voice channels.

- To count up the time: **utils finesse set_property desktop wrapUpCountDown false**
- To count down the time: **utils finesse set_property desktop wrapUpCountDown true**

By default, the value of this property is set to true.

Wrap-Up Button for All Call Types

Use the following CLI command to enable the Wrap-Up button for all call types:

utils finesse set_property desktop enableWrapupButtonForAllCallTypes true

By default, the value of this property is false.

During outbound calls, certain scenarios such as agent-to-agent calls can cause wrap-up operation to fail. However, if this exception scenario does not affect your deployment and you have specific requirements, use this property to enable the **Wrap-Up** button for all call types.

When you use the CLI command **utils finesse set_property desktop enableWrapupButtonForAllCallTypes false** to disable the Wrap-Up button, the button will still be available for the following call types:

- Outbound
- Outbound Callback
- Out

Notification Connection Type

Use the following CLI commands to update the desktop notification connection type as WebSockets or BOSH:

- For WebSockets: **utils finesse set_property desktop notificationConnectionType websocket**
- For BOSH: **utils finesse set_property desktop notificationConnectionType bosh**

By default, the connection type is WebSockets.

Desktop Chat Attachment

Use the following CLI commands to enable or disable the attachment support in Desktop Chat:

- To enable: **utils finesse set_property desktop desktopChatAttachmentEnabled true**
- To disable: **utils finesse set_property desktop desktopChatAttachmentEnabled false**

By default, attachments are enabled in the Desktop Chat.

Desktop Chat Maximum Attachment Size

Use the following CLI commands to configure the attachment size in Desktop Chat:

- **utils finesse set_property desktop desktopChatMaxAttachmentSize** *Attachment Size*

  For example, to set the maximum attachment size to 2 MB, use:

  utils finesse set_property desktop desktopChatMaxAttachmentSize 2097152

Note The maximum attachment size configurable is up to 10 MB.

If you don't configure the maximum attachment size, by default, the maximum attachment size is set to 5 MB.

Desktop Chat Unsupported File Types

The .exe, .msi, .sh, and .bat file types are not supported by default. Use the following CLI commands to override the default list and customize the file types that won't be supported in the Desktop Chat:

- **utils finesse set_property desktop desktopChatUnsupportedFileTypes** *File Types*

  For example, to set the .jar and .bin as unsupported file types, use:

  utils finesse set_property desktop desktopChatUnsupportedFileTypes jar,bin

Multiple file types can be added using a comma-separated string.

Automatic Desktop Login Retries

Cisco Finesse supports automatic desktop login retries when the desktop login fails due to device-related errors. The following properties allow the administrator to control how this feature behaves:

- To enable: **utils finesse set_property desktop enableRetryLoginFeature true**

Attention The **utils finesse set_property desktop enableRetryLoginFeature true** command is not enabling automatic desktop login retries. So, to enable automatic desktop login retries, use the following command:

utils finesse set_property desktop retryLoginAfterLogoutPhoneFailure true

To view the status of automatic desktop login retries, use the following command:

utils finesse show_property desktop retryLoginAfterLogoutPhoneFailure

To disable the automatic desktop login retries, use the following command:

utils finesse set_property desktop enableRetryLoginFeature false

- If this feature is enabled, you can define the retry attempts and intervals.
  - To set the number of retry attempts: **utils finesse set_property desktop loginFailureRetryAttempts <value>**

    The maximum retry attempts are 10. Default value is 3.
  - To set intervals: **utils finesse set_property desktop loginFailureRetryInterval <value>**

    The login retry has a configurable amount of delay between each retry to allow the device to recover. The minimum and maximum interval between retries is 15-180 seconds. Default value is 60 seconds.

Note Reducing the retry interval increases the load on the system when there is a system-wide outage of devices.

By default, the value of this property is set to true.

Sign in as a Mobile Agent

Use the following CLI commands to enable or disable the Sign in as a Mobile Agent feature on the Cisco Finesse sign in page:

- To enable: **utils finesse set_property desktop enableMobileAgentLogin true**
- To disable: **utils finesse set_property desktop enableMobileAgentLogin false**

By default, the value of this property is set to true.

Enable or Disable Keyboard Shortcuts

Use the following CLI commands to enable or disable the keyboard shortcuts for the Cisco Finesse agent and supervisor desktop:

- To enable: **utils finesse set_property desktop enableShortCutKeys true**

- To disable: **utils finesse set_property desktop enableShortCutKeys false**

By default, the value of this property is set to true.

Enable or Disable Drag-and-Drop and Resize for a Gadget or Component

Use the following CLI commands to enable or disable the drag-and-drop and resize features for a gadget or component in the Cisco Finesse desktop:

- To enable: **utils finesse set_property desktop enableDragDropAndResizeGadget true**
- To disable: **utils finesse set_property desktop enableDragDropAndResizeGadget false**

By default, the value of this property is set to false. For more information on using the drag-and-drop and resize features, see the *Cisco Finesse Agent and Supervisor Desktop User Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/finesse/products-user-guide-list.html.

Enable or Disable Preloading of the Secondary Resources

Use the following CLI commands to enable or disable the preloading of the secondary server resources from the alternate side during desktop sign in:

- To enable: **utils finesse set_property desktop preLoadSecondaryResources true**
- To disable: **utils finesse set_property desktop preLoadSecondaryResources false**

The preloaded resources are **images**, **CSS**, **JS**, and **HTML**. The preloading reduces latency and improves performance during desktop failover. By default, the value of this property is set to true.

Security Banner Message for Desktop Users

Cisco Finesse supports custom banner messages in the desktop Sign In page. The administrator defines the banner message for Cisco Finesse desktop users so that they are aware of the security policy while using Cisco Finesse. The banner message can have a maximum of 220 characters. It supports both alphanumeric and special characters. By default, the banner message is not displayed.

- To add the security banner message to the desktop Sign In page: **utils finesse set_property desktop desktopSecurityBannerMessage** <*value*>

  The following example displays the sample security banner that is defined for desktop Sign In page.

```
admin:utils finesse set_property desktop desktopSecurityBannerMessage "IMPORTANT: Finesse
 may only be accessed by authorized users!"

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

No service restart required. Ensure browser is refreshed for the changes to take effect.
```

- To remove the security banner message in the desktop Sign In page: **utils finesse set_property desktop desktopSecurityBannerMessage ""**

Note Cisco Finesse Administration Console and Cisco Finesse desktop now support messages configured in Cisco Unified OS Administration by using the custom logon message feature. From Unified CCX Release 12.5(1)SU1, it's recommended that you use the custom logon message feature as an alternative to the security banner message feature to convey important information to Cisco Finesse desktop users and administrators. For more information about setting up custom logon message, see the *Set Up Customized Logon Message* section in the *Cisco Unified Operating System Administration Guide for Cisco Unified CCX and Cisco Unified IP IVR* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-express/products-maintenance-guides-list.html.

WORK Mode Retention for Non-Voice

Use the following CLI commands to enable or disable the user to remain in WORK mode after CTI reconnection:

- To enable agent to retain WORK mode after CTI reconnection (non-voice): **utils finesse set_property desktop enableAutoWorkModeStateChange false**
- To make the agent available automatically after CTI reconnection (non-voice): **utils finesse set_property desktop enableAutoWorkModeStateChange true**

By default, the value of this property is set to *true* (disabled).

The administrator can enable this CLI, to allow the agents to change to an available state in non-voice MRD explicitly after the Cisco Finesse desktop and media channels are initialized (contrary to the previous behavior where, agents moving automatically to available state, and causing RONA, because of the delay in re-initialization of the gadgets which handle non-voice media).

Dual-Tone Multi-Frequency (DTMF) Desktop Behavior

Note To enable this CLI in Cisco Finesse, install Cisco Finesse 12.5(1)ES2 COP or higher.

The **Wrap-Up** button and the call control buttons, **Hold**, **Transfer**, **Consult**, and **End** are disabled across all calls when DTMF **Keypad** is opened, and until the responses to all DTMF requests are completed or have timed out.

DTMF Pending Requests Threshold Count

When the network or the server is slow to respond, then the response to DTMF requests are delayed. DTMF keypad prevents new operations when more than a configured number of outstanding responses are pending. The default value is 20.

- To configure the DTMF threshold count for pending requests: **utils finesse set_property desktop pendingDTMFThresholdCount** <*value*>

  The following example displays the sample DTMF threshold count.

```
admin:utils finesse set_property desktop pendingDTMFThresholdCount 15

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.
```

```
No service restart required. Ensure the desktop browser is refreshed for the changes
to take effect.
```

DTMF Request Timeout

Cisco Finesse waits for a configured time for each DTMF request. The default timeout is 5 seconds.

- To configure the DTMF timeout for pending requests: **utils finesse set_property desktop dtmfRequestTimeoutInMs** <*value*>

Note The timeout value must be entered in milliseconds.

The following example displays the sample DTMF timeout count.

```
admin:utils finesse set_property desktop dtmfRequestTimeoutInMs 4000

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

No service restart required. Ensure the desktop browser is refreshed for the changes
to take effect.
```

Enable Automatic Device Selection for Single Device

Use the following CLI commands to enable or disable display of **Select Your Preferred Device** window, during agent sign-in, when device selection is enabled for the agent, and the extension is associated with only one active device at the time of sign-in:

- To enable: **utils finesse set_property desktop enableDeviceSelectionForSingleDevice** *true*
- To disable: **utils finesse set_property desktop enableDeviceSelectionForSingleDevice** *false*

Note This property is applicable only when there's a single active device available for the extension at the time of sign-in. However, the device selection window comes up when there's no active device, and when there are multiple active devices for the extension.

Cisco Finesse implicitly selects one device as agent's active device when the property is set to **False**. If the value is set to **True**, then the device selection window appears. By default, the value is set to **True** for Unified CCE deployments.

Maximum Number of Visible Multi-Tab Gadget Tabs

The **utils finesse set_property desktop maxTabsOnTabbedGadgetHeader** CLI command is used to configure the maximum number of gadgets that appear on the Multi-Tab gadget header. Administrators can customize the **maxTabsOnTabbedGadgetHeader** property to display more gadget tabs in the Multi-Tab gadget header before they are moved to the gadget drop-down. There are no restrictions on the maximum number of gadget tabs that appear on the Multi-Tab gadget header. The default value is set to seven.

Use the following commands to view and set the maximum number of tabs that appear in the Multi-Tab gadget header:

- **utils finesse show_property desktop maxTabsOnTabbedGadgetHeader**: This command displays the maximum number of tabs that are configured to be displayed in the header of Multi-Tab gadgets. The default value is seven.

  Example

  ```
  admin:utils finesse show_property desktop maxTabsOnTabbedGadgetHeader
  The value of property 'maxTabsOnTabbedGadgetHeader' is '7'
  ```

- **utils finesse set_property desktop maxTabsOnTabbedGadgetHeader** <*value*>: This command sets the maximum number of tabs that are displayed in the header of the Multi-Tab gadgets.

  Example

  ```
  admin:utils finesse set_property desktop maxTabsOnTabbedGadgetHeader 5
  Property successfully updated.
  Ensure property is updated in all Finesse nodes in the cluster.
  No service restart required.
  Ensure the desktop browser is refreshed for the changes to take effect.
  ```

Note This command is used to set the maximum number of gadgets that can be displayed in the header. If you have configured more gadgets for a Multi-tab gadget, the remaining gadgets are listed in the drop-down.

Non-Page Level Gadgets Preceding Page-Level Gadgets in a Multi-Tab Gadget

A Multi-Tab Gadget can consist of both page-level and non page-level gadgets. In this case, page-level gadgets appear first in sequence in the Multi-Tab header by default. However, you can use the following CLI commands to allow non page-level gadgets to appear before the page-level gadgets in a Multi-tab gadget header:

- **utils finesse show_property desktop displayNavLevelGadgetsFirstInMultitab**

  The **utils finesse show_property desktop displayNavLevelGadgetsFirstInMultitab** command allows you to check if non page-level gadgets are displayed first in the Multi-Tab gadget header. If the result of the command is **true**, the non page-level gadgets are displayed first in sequence. If the result of the command is **false**, the page-level gadgets are displayed first in sequence.

  Example

  ```
  admin:utils finesse show_property desktop displayNavLevelGadgetsFirstInMultitab
  The value of property 'displayNavLevelGadgetsFirstInMultitab' is 'true'
  ```

- **utils finesse set_property desktop displayNavLevelGadgetsFirstInMultitab <value>**

  The **utils finesse set_property desktop displayNavLevelGadgetsFirstInMultitab** command configures the order of non page-level gadgets and page-level gadgets in a Multi-Tab gadget header. If the command is set to **true**, the non page-level gadgets are displayed first in sequence. If the command is set to **false**, the page-level gadgets are displayed first in sequence.

  Example

  ```
  admin:utils finesse set_property desktop displayNavLevelGadgetsFirstInMultitab true
  Property successfully updated.
  Ensure property is updated in all Finesse nodes in the cluster.
  No service restart required. Ensure the desktop browser is refreshed for the changes
  to take effect.
  ```

Notifications from Call Control in Multi-Tab Gadgets

Use the following CLI commands to configure the notification settings for Call Control gadget tab when it is configured as a tab within a Multi-Tab gadget:

- **utils finesse set_property desktop showNotificationOnCallVariablesChange** *true*

  The **utils finesse set_property desktop showNotificationOnCallVariablesChange** command changes the notification settings when the call variables are updated.

  Example

  ```
  admin:utils finesse set_property desktop showNotificationOnCallVariablesChange true
  Property successfully updated.
  Ensure property is updated in all Finesse nodes in the cluster.
  No service restart required. Ensure the desktop browser is refreshed for the changes
  to take effect.
  ```

- **utils finesse set_property desktop showNotificationOnParticipantChange** *true*

  The **utils finesse set_property desktop showNotificationOnParticipantChange** command changes the notification settings when participants in the call change.

  Example

  ```
  admin:utils finesse set_property desktop showNotificationOnParticipantChange true
  Property successfully updated.
  Ensure property is updated in all Finesse nodes in the cluster.
  No service restart required. Ensure the desktop browser is refreshed for the changes
  to take effect.
  ```

- **utils finesse set_property desktop showNotificationOnWrapUpReasonsChange** *true*

  The **utils finesse set_property desktop showNotificationOnWrapUpReasonsChange** command changes the notification settings whenever the wrap-up reasons for the call change. This command is applicable to a call with multiple participants, and the notification appears for all agents excluding the agent who changed the wrapUpReason. So, the notification does not appear for a call with a single agent.

  Example

  ```
  admin:utils finesse set_property desktop showNotificationOnWrapUpReasonsChange true
  Property successfully updated.
  Ensure property is updated in all Finesse nodes in the cluster.
  No service restart required. Ensure the desktop browser is refreshed for the changes
  to take effect.
  ```

- **utils finesse set_property desktop showNotificationOnNumCallsChange** *true*

  The **utils finesse set_property desktop showNotificationOnNumCallsChange** command changes the notification settings when the number of calls change for an agent.

  Example

  ```
  admin:utils finesse set_property desktop showNotificationOnNumCallsChange true
  Property successfully updated.
  Ensure property is updated in all Finesse nodes in the cluster.
  No service restart required. Ensure the desktop browser is refreshed for the changes
  to take effect.
  ```

WebProxy Service

WebProxy Service acts as a transparent reverse proxy between external clients and the Finesse service. It provides SSL termination and caching services to the Finesse server to reduce latency and improve performance.

Configuration changes done on the Finesse server may not be immediately available to the clients due to the intermediary webproxy cache. The administrator can clear the intermediary webproxy cache using **utils webproxy cache clear**.

WebProxy cache is automatically cleared when you restart the Finesse Tomcat service. Static resources (images and scripts), Shindig gadget XML, and resources are cached until the Finesse Tomcat service is restarted or explicitly cleared by the administrator.

For more information on REST API Response Caching, see *REST API Developer Guide* at https://developer.cisco.com/docs/finesse/.

The logging level of the WebProxy Service is managed using the web proxy log-levels CLI.

Note WebProxy Service CLIs are node-specific and must be run on all nodes in the cluster.

Proxy cache bypassing reduces performance and must be used for debugging purposes during the gadget development or troubleshooting.

Server cache for the Finesse API can be bypassed by including `bypassServerCache=true` as a query parameter in the request or clear server cache using **utils webproxy cache clear**.

Server cache for the Finesse desktop can be bypassed by including `bypassServerCache=true&nocache` as a query parameter in the desktop URL.

utils webproxy cache clear

This command clears the cache from the WebProxy Service.

Command Syntax

utils webproxy cache clear {*all* | *webproxy* | *desktop* | *rest* | *shindig*}

Options

- *all*—Clears all the configured caches.
- *webproxy*—Clears the default webproxy cache.
- *desktop*—Clears the desktop cache. The desktop cache contains static HTML, CSS, scripts, and icons used in the Finesse desktop.
- *rest*—Clears the REST APIs cache. The REST API responses cached are:
  - ECCVariableConfig
  - MediaDomain
  - TeamResource APIs include ReasonCodes, WrapUpReasons, MediaPropertiesLayouts, PhoneBooks, and WorkFlows. The responses of the TeamResource API are cached at the team-level.

- *shindig*—Clears the Shindig cache. The Shindig cache contains XML gadget definition (ifr request-response) and gadget resources (concat request-response).
- *authmode*—Clears the UserAuthMode API cache.

Command Modes

Administrator (admin)

Requirements

Command privilege level: 1

Allowed during upgrade: Yes

Applies to: Unified CCE, Unified CCX, and Packaged CCE

Example

```
admin:utils webproxy cache clear desktop
Successfully cleared desktop cache
```

set webproxy access-log-level

This command sets the log-level for the access logs generated by the WebProxy Service. The access logs record information about all external requests that reach the proxy. The requests are logged in the access log after the request is processed.

Command Syntax

set webproxy access-log-level {*off* | *info* | *debug*}

Options

- *off*—Turns off the logging into the access logs of the WebProxy Service.
- *info*—Sets the log-level for access logs of the WebProxy Service to information. This captures the data of each request such as time, client, host, user, and so on.
- *debug*—Sets the log-level for access logs of the WebProxy Service to debug. This captures the detailed data of each request for debugging.

Note Setting the access logs to debug impacts performance. Hence, avoid using in the production deployment.

Command Default

The default value is *off*.

Command Modes

Administrator (admin)

Requirements

Command privilege level: 1

Allowed during upgrade: Yes

Applies to: Unified CCE, Unified CCX, and Packaged CCE

Example

```
admin:set webproxy access-log-level off
Webproxy access log-level is turned off

admin:set webproxy access-log-level info
Successfully set webproxy access log-level to info
Service restarted
```

set webproxy log-severity

This command sets the severity of the error logs that are generated by the WebProxy Service. The error logs record information about encountered issues of different severity levels.

Command Syntax

set webproxy log-severity {*debug* | *warn* | *error* | *crit* | *alert* | *emerg*}

Options

- *debug*—Sets the severity level to debug.

Note Setting the error logs to debug impacts performance. Hence, avoid using in the production deployment.

- *warn*—Sets the severity level to warning.
- *error*—Sets the severity level to error.
- *crit*—Sets the severity level to critical.
- *alert*—Sets the severity level to alert.
- *emerg*—Sets the severity level to emergency.

Command Default

The default value is *warn*.

Command Modes

Administrator (admin)

Requirements

Command privilege level: 1

Allowed during upgrade: Yes

Applies to: Unified CCE, Unified CCX, and Packaged CCE

Example

```
admin:set webproxy log-severity warn
Successfully set webproxy log severity to warn
Service restarted
```

show webproxy access-log-level

This command displays the configured log-level for the access logs of the WebProxy Service.

Command Syntax

show webproxy access-log-level

Command Modes

Administrator (admin)

Requirements

Command privilege level: 1

Allowed during upgrade: Yes

Applies to: Unified CCE, Unified CCX, and Packaged CCE

Example

```
admin:show webproxy access-log-level
Current webproxy access log-level is: info
```

show webproxy log-severity

This command displays the configured severity level for the error logs of the WebProxy Service.

show webproxy log-severity

Command Modes

Administrator (admin)

Requirements:

Command privilege level: 1

Allowed during upgrade: Yes

Applies to: Unified CCE, Unified CCX, and Packaged CCE

Example

```
admin:show webproxy log-severity
Current webproxy log-severity is: warn
```

Service Properties

Configure the service properties using the following CLIs for the features.

Note The CLIs require Cisco Finesse Tomcat restart except for desktop related properties.

Security Banner Message for Administrators

Cisco Finesse supports custom banner messages in the administration Sign In page. The administrator defines the banner message for the users so that they are aware of the security policy while using Cisco Finesse. The banner message can have a maximum of 220 characters. It supports both alphanumeric and special characters. By default, the banner message is not displayed.

- To add the security banner message to the administrator Sign In page: **utils finesse set_property admin adminSecurityBannerMessage <value>**

The following example displays the sample security banner that is defined for the administrator Sign In page.

```
admin:utils finesse set_property admin adminSecurityBannerMessage "IMPORTANT: Finesse may
only be accessed by authorized users!"

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat Service for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

- To remove the security banner message in the administrator Sign In page: **utils finesse set_property admin adminSecurityBannerMessage ""**

Note Cisco Finesse Administration Console and Finesse desktop now support messages configured in Cisco Unified OS Administration by using the custom logon message feature. For more information about setting up a custom logon message, see the *Set Up Customized Logon Message* section in the *Cisco Unified Operating System Administration Guide for Cisco Unified CCX and Cisco Unified IP IVR* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-express/products-maintenance-guides-list.html.

Enable or Disable User Authentication Discovery API

Use the following CLI commands to enable or disable the `UserAuthMode` API. This API allows a client to discover the authentication mode of a user in Unified CCE deployments when the system is in hybrid mode (SSO or non-SSO). By default, this API is enabled.

Note This API does not require HTTP BASIC authentication. It is provided for third-party integration to decide if the user authentication must proceed with SSO or non-SSO authentication modes.

- To enable: **utils finesse set_property webservices enableUserAuthMode true**
- To disable: **utils finesse set_property webservices enableUserAuthMode false**

Enable or Disable Plain XMPP Socket—Port 5222

Use the following CLI commands to enable or disable the Cisco Finesse Notification Service plain XMPP port (5222). This port can be enabled only if you have third-party solutions that connect directly to the Cisco Finesse Notification Service over plain Transmission Control Protocol (TCP) connection. This port is not required for the Finesse desktop or BOSH/WebSocket based integrations. By default, the port is disabled.

- To enable: **utils finesse set_property webservices enableInsecureOpenfirePort true**
- To disable: **utils finesse set_property webservices enableInsecureOpenfirePort false**

Enable or Disable Secure XMPP Socket—Port 5223

Use the following CLI commands to enable or disable the external access to the Cisco Finesse Notification Service TCP-based XMPP port (5223). The port must be enabled for external client connectivity only if you have third-party solutions that connect directly to the Cisco Finesse Notification Service over this port. By default, the port is enabled (value is set to *true*).

When the port is enabled, it can be accessed by the Cisco Finesse nodes (primary and secondary) and by external clients. When the port is disabled, it cannot be accessed by external clients.

- To enable: **utils finesse set_property webservices enableExternalNotificationPortAccess true**
- To disable: **utils finesse set_property webservices enableExternalNotificationPortAccess false**

Note Restart Cisco Finesse Tomcat and Cisco Finesse Notification Services for the changes to take effect.

Restricting Access to the External XMPP Notification Port 5223

Use the following CLI commands to restrict the IP addresses from accessing the TCP-based XMPP notification port (5223) available for external client connectivity. You can add, delete, or view the configured IP addresses only when the **enableExternalNotificationPortAccess** property is enabled on all the Finesse nodes in the cluster.

Note These restrictions do not affect the desktop XMPP notification port 7443.

To enable access to port 5223 use the CLI command **utils finesse set_property webservices enableExternalNotificationPortAccess true**.

- **utils finesse notification external_port_access add** *[ip1,ip2,ip3]*—This command adds one or more IP addresses to the list of hosts that are configured to access Cisco Finesse XMPP notification port 5223. Multiple IP addresses can be provided as a comma-separated list. Wildcard character * is not supported.

  Example

  ```
  admin:utils finesse notification external_port_access add 10.10.10.21,10.10.255.25

  Successfully added 2 IP address(es). Ensure that the IP address(es) are added,
  and verify that external notification port access is enabled in all the Finesse nodes
  in the cluster.
  Please refer to 'utils finesse show_property webservices
  enableExternalNotificationPortAccess'.

  Restart Cisco Finesse Notification Service for the changes to take effect:
  utils service restart Cisco Finesse Notification Service
  ```

- **utils finesse notification external_port_access delete**—This command deletes one or more IP addresses from the list of hosts that are configured to access Cisco Finesse XMPP notification port 5223. Multiple IP addresses can be provided as a comma-separated list.

  Example

  ```
  admin:utils finesse notification external_port_access delete 10.10.10.21,10.10.255.25

  Successfully deleted 2 IP address(es). Verify that the IP address(es) are deleted in
  ```

```
all the Finesse nodes in the cluster.

Restart Cisco Finesse Notification Service for the changes to take effect:
utils service restart Cisco Finesse Notification Service
```

- **utils finesse notification external_port_access delete_all**—This command deletes all the configured IP addresses allowed to access the Cisco Finesse XMPP notification port 5223.

  Example

```
admin:utils finesse notification external_port_access delete_all

Do you want to delete all IP address(es) (y/n): y

Successfully deleted all IP address(es). Verify that the IP address(es) are deleted in
 all the Finesse nodes in the cluster.

Restart Cisco Finesse Notification Service for the changes to take effect:
utils service restart Cisco Finesse Notification Service
```

- **utils finesse notification external_port_access list**—This command lists all the configured IP addresses allowed to access the Cisco Finesse XMPP notification port 5223.

  Example

```
admin:utils finesse notification external_port_access list

The following IP address(es) are configured to access the notification port:
10.10.10.21
10.10.255.25

External notification port access is disabled in the present node. Verify that is enabled
 in all the Finesse nodes in the cluster.
Please refer to 'utils finesse show_property webservices
enableExternalNotificationPortAccess'.
```

Enable or Disable Enforcement of X.509 Certificate Trust Validation

Use the following CLI commands to enable or disable the validation of the X.509 CA or the selfsigned certificate. From Release 12.5(1) onwards, Cisco Finesse validates SSL certificates of all the servers (CUCM and Customer Collaboration Platform) it communicates. This requires the custom CA providers or the selfsigned certificates of the server it communicates to be present in the Cisco Finesse Tomcat trust store. If the certificates are not added into the Cisco Finesse trust store, then certain interactions can fail. It is advised to add the certificates into the Cisco Finesse trust store. If any user chooses to ignore the validation, enforcement can be turned off. This CLI allows users to disable or enable validation. By default, the validation is turned on.

- To enable: **utils finesse set_property webservices trustAllCertificates true**
- To disable: **utils finesse set_property webservices trustAllCertificates false**

Enable or Disable Call Variables Logging

Use the following CLI commands to enable or disable the call variables logging. The callVariables contain sensitive user information and this property allows the administrator to decide whether the information must be captured in the logs. By default the property is disabled.

- To enable:

  utils finesse set_property webservices logCallVariables true

  utils finesse set_property fippa logCallVariables true

- To disable:

  utils finesse set_property webservices logCallVariables false

  utils finesse set_property fippa logCallVariables false

Permissions to Drop Participants from Conference

Note To enable this CLI in Cisco Finesse, install Finesse 12.5(1) ES3 COP or higher.

Use the following commands to allow an agent or a supervisor, who is the participant in a conference call, to drop another agent, supervisor, or caller (participants) from the conference call.

Note Only agents and supervisors can drop participants in the conference call.

- **utils finesse set_property webservices enableDropParticipantFor supervisor_only**—This command allows only the supervisor, who is a participant of the conference call, to drop other agents in the conference call. The supervisor cannot drop a CTI Route Point, IVR port, a device to which no agent is signed in, or a caller device. By default, this property is set to **supervisor_only**.
- **utils finesse set_property webservices enableDropParticipantFor conference_controller_and_supervisor**—This command allows,
  - the supervisor to drop any agents, CTI Route Point, IVR port, a device to which no agent is signed in, or a caller device in the conference call.
  - the conference controller (an agent who initiated the conference call) to drop another agent, supervisor, CTI Route Point, IVR port, a device to which no agent is signed in, or a caller device in the conference call.

Note To enable the supervisor or call controller to drop an unmonitored extension in Cisco Unified CCE, in Release 12.0(1) or higher, set the **DropAnyPartyEnabled** registry key to *1* in the Dynamic Registry of the CTI server. The supervisor cannot drop a CTI Route Point, IVR port, a device to which no agent is signed in, a caller device, or other agents for whom SILENT_MONITOR is not initiated by the supervisor.

For more information, see the *Enable Dropping Call Participants from a Conference Call* section in *Cisco Contact Center Gateway Deployment Guide for Cisco Unified ICM/CCE* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-programming-reference-guides-list.html.

- **utils finesse set_property webservices enableDropParticipantFor all**—This command allows any agent or supervisor in the conference call to drop another agent, supervisor or the caller. To ensure that this feature works properly on Finesse desktop, you must update the **enableDropParticipantFor** value

for desktop also. For more information on enabling the desktop property, refer to Drop Participants from Conference.

Enable Team API Access for All Users

A new configuration property `enableTeamAPIAccessForAllusers` is added for enabling the Team API access for all agents and supervisors. When you enable this property, all agents and supervisors will be able to access information of all the teams without any restriction.

If this property is disabled, only administrator and supervisors can access the Team API. Supervisors can access the information of the teams that they are assigned to and Administrators can access all the teams. By default this property is disabled.

Use the following command to enable this configuration property:

utils finesse set_property webservices enableTeamAPIAccessForAllusers true.

Example:

```
admin:utils finesse set_property webservices enableTeamAPIAccessForAllusers true

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat Service for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

Use the following command to disable this configuration property:

utils finesse set_property webservices enableTeamAPIAccessForAllusers false

Example:

```
admin:utils finesse set_property webservices enableTeamAPIAccessForAllusers false

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat Service for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

Log Collection Schedule

Use the following CLIs to create, list, and delete automatic desktop log collection schedules for agents and supervisors. This can also be used for debugging purposes.

utils finesse desktop_auto_log_collection create: This command creates a schedule that collects the agent's browser logs. You can create up to five log collection schedules for up to 15 agents.

While creating the log schedule, specify the agent IDs, log collection interval, and duration up to when the logs are to be collected.

The log collection interval and the duration have to be between 30 to 900 seconds. The logs that are collected during the schedule are received in a .zip file format. The logs are collected at: `/opt/cisco/desktop/logs/clientlogs`.

Example:

```
admin:utils finesse desktop_auto_log_collection create

Initializing command line interface...
Checking Cisco Finesse Tomcat status...

Enter agent IDs to continue. (Maximum 15 agents) [Example : 1001001,1001002] : 1001002
Agent IDs entered: 1001002
Enter duration in seconds.(value between 30 and 900) : 240
Duration entered: 240
Enter interval in seconds.(value between 30 and 240) : 60
Interval entered: 60

Successfully scheduled client log collection for the specified agent(s).

Ensure the same is enabled in all the Finesse nodes in the cluster..
```

utils finesse desktop_auto_log_collection list: This command lists all active log collection schedules.

Example:

```
admin:utils finesse desktop_auto_log_collection list

Initializing command line interface...
Checking Cisco Finesse Tomcat status...
These are the live log collection schedules:

Schedule ID:1 Created At: Thu Jun 6 23:23:53 PDT 2019
Duration: 240 seconds
Frequency: 60 seconds
Agent Ids: 1001002
```

utils finesse desktop_auto_log_collection delete: This command deletes the active log collection schedules. When this command is run, all the active log collection schedules are displayed and you are prompted to enter the Schedule ID that you want to delete.

Example:

```
admin:utils finesse desktop_auto_log_collection delete

Initializing command line interface...
Checking Cisco Finesse Tomcat status...
These are the live log collection schedules:

Schedule ID:1 Created At: Thu Jun 6 23:23:53 PDT 2019
Duration: 240 seconds
Frequency: 60 seconds
Agent Ids: 1001002
Enter schedule ID to delete (enter 'all' to delete all): 1
Schedule ID entered: 1

Successfully deleted the log collection with schedule id : 1
```

Upgrade

Upgrade-related commands are grouped under **utils system upgrade**.

utils system upgrade initate: This command allows you to initiate and install upgrades and Cisco Option Package (COP) files from both local and remote directories.

utils system upgrade cancel: This command allows you to cancel an upgrade.

utils finesse layout updateCuicGadgetUrl: The command has the following options:

- **utils finesse layout updateCuicGadgetUrl 12.5.1**—This command allows you to change the .jsp references of Cisco Unified Intelligence Center (CUIC) gadgets to .xml with no functional changes in the Finesse desktop layout. This command updates the CUIC gadgets URL path to work with CUIC, Release 12.5(1). The changes are applicable for both Finesse default desktop layout and team desktop layouts.

Note

- The administrator must run the CLI on the primary Finesse server.
- The changes are not reversible. The changes are not compatible with CUIC releases prior to 12.5(1).

```
admin:utils finesse layout updateCuicGadgetUrl 12.5.1
This command will update the CUIC URI configured in the desktop layouts to work with
CUIC version 12.5.1.The changes are applied to both default and team layouts.
*WARNING* - The changes are not reversible & are not compatible with older CUIC
versions.Please ensure that you have the matching CUIC version.
Do you wish to continue [ y / n] ?yes

All CUIC URIs have been modified in layouts.

Restart Cisco Finesse Tomcat for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

- **utils finesse layout updateCuicGadgetUrl 12.6.1+**—This command allows you to change the .jsp references of CUIC to .xml with no functional changes in the Finesse desktop layout. This command updates the CUIC gadgets URL path to work with CUIC, Release 12.6(1) and later versions. The changes are applicable for both Finesse default desktop layout and team desktop layouts.

Note

- The administrator must run the CLI on the primary Finesse server.
- The changes are not reversible. The changes are not compatible with CUIC releases prior to 12.5(1).

```
admin:utils finesse layout updateCuicGadgetUrl 12.6.1+
This command will update the CUIC URI configured in the desktop layouts to work with
CUIC version 12.6 & later versions.The changes are applied to both default and team
layouts.
*WARNING* - The changes are not reversible & are not compatible with older CUIC
versions.Please ensure that you have the matching CUIC version.

Do you wish to continue [ y / n] ?y

All CUIC URIs have been modified in layouts.

Restart Cisco Finesse Tomcat for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

Cisco Finesse Release 12.5(1) onwards, CUIC supports only XML gadgets. Switching to XML-based gadgets reduces latency and improves performance. After the installation of CUIC or Co-resident deployment, run this command to optimize the loading of CUIC gadgets.

For CUIC Release 12.5(1) gadgets (Live Data and Historical) to load in Cisco Finesse, the administrator must enable CORS on CUIC server using the **utils cuic cors enable** command.

For more information, see *Administration Console User Guide for Cisco Unified Intelligence Center* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-maintenance-guides-list.html.

Shutdown

Use the following command to shut down Finesse:

utils system shutdown

If the virtual hosts running the Finesse servers are also shut down during a maintenance event, to power up Finesse after the maintenance event is complete, you must sign in to the ESXi host or its vCenter with vSphere Client and power up the virtual machines for primary and secondary Finesse servers.

Replication Status

To check replication status, run the following command on the *primary* Finesse server:

- **utils dbreplication runtimestate**

  This command returns the replication status on primary and secondary Finesse servers.

- Check the RTMT counter value for replication. If all nodes in the cluster show a replication status of 2, replication is functioning correctly.
- If the RTMT counter value for replication status is 3 or 4 for all nodes in the cluster, replication is set up but an error occurred and replication is not functioning properly.
- If the majority of the nodes show a value of 0 or 1, run the command **utils dbreplication reset all** from the primary Finesse server.
- If any node shows any replication value other than 1 or 2, replication is not set up correctly.
- To fix replication, contact Cisco Technical Support.

Note The DB replication setup must be completed to reflect the following primary node changes to the secondary node.

- Reason Codes
- Wrap-Up Reasons
- Media Properties Layouts
- Phone Books
- Workflows
- Team Message

View Property

Use the following CLIs to view the property values across all property files.

- **utils finesse show_property fippa property_name**: To view the specified Finesse IPPA property's value.
- **utils finesse show_property desktop property_name**: To view the specified desktop property's value.
- **utils finesse show_property webservices property_name**: To view the specified web service property's value.
- **utils finesse show_property admin securityBannerMessage**: To view the specified banner message for the administrator Sign In page.

Note The View property CLIs do not support multiple values.

Update Property

Use the following CLIs to update the property values across all property files.

- **utils finesse set_property desktop property_name property_value**: To update an existing property value used by the Finesse desktop service.
- **utils finesse set_property fippa property_name property_value**: To update an existing property value used by the Finesse IPPA service.
- **utils finesse set_property webservices property_name property_value**: To update an existing property value used by the Finesse web service.
- **utils finesse set_property admin adminSecurityBannerMessage**: To update an existing property value used by the Finesse administrator for the security banner message.

Signout from Media Channels

The CLI **utils finesse user_signout_channel** is used by the Administrator to configure the media channels from which the users are signed out.

When signing out from Cisco Finesse, the CLI **utils finesse user_signout_channel type** lists all the choices of media channels from which the user is signed out. For example:

utils finesse user_signout_channel type

1: signout user from voice channel.

2: signout user from voice and non-voice media channels configured for Cisco Finesse.

a: signout from all media channels configured for the user.

Note This is default behavior. It is suitable if the non-voice media is running as a gadget within Finesse Desktop and hence, it is valid to assume that the desktop user cannot handle tasks when signing out of Finesse.

q: quit.

Select the choice of media [1-2 or a,q]: 2

User signout channel type is now changed to "signout user from voice and non-voice media channels configured for Cisco Finesse."

Note **user_signout_channel type** must be updated for all Cisco Finesse nodes in the cluster.

For any changes done to media channels, it will take fifteen minutes for the new media channels signout to take effect.

The CLI **utils finesse user_signout_channel status** displays the type of media channels from which the user is signed out.

Finesse Maintenance Mode Services

Use the following CLI commands to start and check the status of the Finesse Maintenance Mode on the Cisco Finesse server.

utils finesse maintenance initiate

Initiates the maintenance mode on the Cisco Finesse server from the command line.

utils finesse maintenance status

Checks the status of the maintenance mode of the Cisco Finesse server from the command line.

ConnectedUsersInfo

Use the following CLI command to view the list of users connected to the Cisco Finesse server where the CLI is run.

utils finesse show_connected_users summary

Provides the summary information about the connected users in the Cisco Finesse server where the CLI is run.

If the above command is run, it lists the total number of users connected to the Cisco Finesse server where the CLI is run along with the number of users connected through Cisco Finesse Desktop, Finesse IP Phone, and third-party desktops.

Example is as follows:

```
admin: utils finesse show_connected_users summary
Total Connected Users: 2
Desktop Users: 2
FIPPA Users: 0
```

```
Third-party Users: 0
Users connected to Finesse via LAN/WAN: 1
Users connected to Finesse via Proxy: 1
To view the complete list of signed-in users, log in to the Cisco Finesse
Administration Console, and navigate to the Connected Agents tab.
```

set webapp session maxlimit

This command sets the maximum limit for the number of concurrent Unified CCX web application sessions per user.

For the new setting to become effective, you must restart the node. Until you restart the node, the system continues to use the old values. In a HA setup, you must run this command on both the nodes. This command prompts you to restart the node.

Note Restart the nodes during off-peak traffic hours to avoid impact on the system performance.

This setting is preserved during software upgrades on both the nodes.

If the number of sessions is limited to 1 on both nodes, a user is allowed to have one session each on both the nodes.

Command syntax

set webapp session maxlimit *number*

Syntax Description

Parameters	Description
number	Specifies the number to limit the concurrent web application sessions. The value ranges from 1 to 10. Default value is 10. **Note** When you exceed the defined limit for maximum number of signed in sessions, the interface sign-in page displays the Logon Status message as: `The Session limit has already been reached for <username>. Please logout from those sessions or wait <Value> minutes for inactive sessions to be automatically closed.`

Command Modes

Administrator

Requirements

Command privilege level: 1

Allowed during upgrade: No

Applies to: Cisco Identity Service Management, Disaster Recovery System, Cisco Unified CCX Administration, Cisco Finesse Administration, Cisco Unified Serviceability, Cisco Unified CCX Serviceability, Cisco Unified OS Administration, and Cisco Unified Intelligence Center.

Example

```
admin:set webapp session maxlimit 4

**********************************W A R N I N
G**********************************
The node needs to be restarted for the changes to take effect.This will
disconnect active web sessions and all web applications on this node will be
unavailable
until the node restarts.This node restart will take several minutes to complete.
Do you want to continue (yes/no) ? yes

*************Restarting node***************

The system is going down for reboot in 1 Minute
The webapp session limit has been successfully set to 4.
```

set webapp session timeout

This command sets the time in minutes to invalidate any inactive Unified CCX web application sessions. After the set time elapses, the users are logged off from any of the inactive Unified CCX web sessions. The default session timeout value is 30 minutes.

For the new setting to become effective, you must restart the node. Until you restart the node, the system continues to use the old values. In a HA setup, you must run this command on both the nodes. This command prompts you to restart the node.

Note Restart the nodes during off-peak traffic hours to avoid impact on the system performance.

This setting is preserved during software upgrades on both the nodes.

Command syntax

set webapp session timeout *minutes*

Syntax Description

Parameters	Description
minutes	Specifies the time, in minutes, that must elapse before a web application times out and logs off the user. • Value range: 5-35000 minutes • Default value: 30 minutes

Command Modes

Administrator

Requirements

Command privilege level: 1

Allowed during upgrade: No

Applies to: Cisco Identity Service Management, Disaster Recovery System, Cisco Unified CCX Administration, Cisco Finesse Administration, Cisco Unified Serviceability, Cisco Unified CCX Serviceability, and Cisco Unified OS Administration.

Example

```
admin:set webapp session timeout 20
Continuing with this operation will set the session-timeout for web sessions to
20 minutes
after the node has been rebooted.
Continue (y/n)?y
web session-timeout updated to 20 minutes.

The node has to be rebooted for the changes to take effect immediately.
This will disconnect active web sessions.
Continue (y/n)?n
The updated web session time-out would take effect on next reboot

The current session-timeout used for web sessions and applications is 30 minutes.
The updated session-timeout value of 20 minutes will take effect on restart of
the node.
```

Update Cloud Connect Connection Time

When there is a low bandwidth, the default value that is used for an HTTP client for obtaining the Cloud Connect token may not be sufficient and result in timeout. The following CLI commands can be used to view and update the connection timeout values. The default value for **cloudconnectHttpConnectionTimeout** is 5000 milliseconds and the default value for **cloudconnectHttpReadTimeout** is 10000 milliseconds.

To view the current values, use the following commands:

utils finesse show_property webservices cloudconnectHttpConnectionTimeout

utils finesse show_property webservices cloudconnectHttpReadTimeout

To update the values, use the following commands:

utils finesse set_property webservices cloudconnectHttpConnectionTimeout
<time_value_in_milliseconds>

utils finesse set_property webservices cloudconnectHttpReadTimeout
<time_value_in_milliseconds>

For example, the **utils finesse set_property webservices cloudconnectHttpConnectionTimeout** 8000 command updates the HTTP connection timeout value to 8000 milliseconds.

The **utils finesse set_property webservices cloudconnectHttpReadTimeout** 12000 command updates the HTTP connection read timeout value to 12000 milliseconds.

AI Services Configuration

Finesse supports configuring AI services that are available through Agent Answers, Call Transcript, and Recording gadgets. By default, the AI services are disabled in Finesse. Use the following CLI commands to enable, disable, or view the status of the AI services. To enable or disable the AI services, the commands must be run in all the Finesse clusters.

Note Enable the AI services only if you are using Unified CCE 12.5(1). For Unified CCE 12.6(1) and above, this service is available without enabling.

Use the following CLI command to enable AI services:

utils finesse set_property webservices enableCustomAgentServices true

Example:

```
admin:utils finesse set_property webservices enableCustomAgentServices true

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat Service for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

Use the following CLI command to disable AI services:

utils finesse set_property webservices enableCustomAgentServices false

Example:

```
admin:utils finesse set_property webservices enableCustomAgentServices false

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat Service for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

Use the following CLI command to view the status of the AI services:

utils finesse show_property webservices enableCustomAgentServices

Example:

```
admin:utils finesse show_property webservices enableCustomAgentServices

 The value of property 'enableCustomAgentServices' is 'false'
```

Note If **enableCustomAgentServices** is enabled in Finesse, it will take the precedence over the configuration done in Cisco Unified CCE 12.6(1).

The configured AI services can also be managed through Finesse CLI commands. By default, AI services available through Agent Answers and Call Transcript gadgets are configured.

Use the following CLI command to modify configured AI services:

utils finesse set_property webservices customAgentService <values>

The parameter **values** allows comma separated numeric values ranging from 1 to 3. One or multiple values can be entered.

- 1—Agent Answers
- 2—Call Transcript
- 3—Recording

To enable only the Agent Answers gadget, use the following CLI:

utils finesse set_property webservices customAgentServices 1

Example 1:

```
admin:utils finesse set_property webservices customAgentServices 1

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat Service for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

To enable all the AI gadgets, use the following CLI:

utils finesse set_property webservices customAgentServices 1,2,3

Example 2:

```
admin:utils finesse set_property webservices customAgentServices 1,2,3

Property successfully updated.
Ensure property is updated in all Finesse nodes in the cluster.

Restart Cisco Finesse Tomcat Service for the changes to take effect:
utils service restart Cisco Finesse Tomcat
```

Certificate Configuration

show tls server cert_type

This command displays the configured certificate type used by the server for TLS connections.

Command syntax

show tls server cert_type

Requirements

Level privilege: 1

Command privilege level: 1

Allowed during upgrade: No

```
admin:show tls server cert_type
The server certificate type is set to ECDSA
```

```
Command successful
```

set tls server cert_type

Use this command to set the server certificate type to either RSA or ECDSA ciphers for TLS connections. The certificate set to the specified type is then presented for the cipher negotiations on all incoming TLS communications.

Command syntax

set tls server cert_type [option]

Option

ecdsa—Sets the certificate type to ECDSA.

rsa—Sets the certificate type to RSA.

Requirements

Level privilege: 1

Command privilege level: 1

Allowed during upgrade: No

Example

```
admin:set tls server cert_type rsa

Configuring the server to use RSA certificates for all inbound connections.

Do you want to continue (y/n) ? y
Yes entered
Configuring the server to use RSA ciphers for inbound connections.

Successfully configured the server to use RSA certificate for all inbound
connections.

******************************************************
A system reboot will occur for the changes to take effect.
It is highly recommended that you perform a system backup
after the system reboot.
Ensure all the nodes in the cluster are running on the same
certificate type by running the 'set' command
******************************************************

Broadcast message from root@uccxfirstnode (Mon Jul 5 10:31:04 2021):

The system is going down for reboot in 1 Minute

Broadcast message from root@uccxfirstnode (Mon 2021-07-05 10:31:05 IST):

The system is going down for reboot at Mon 2021-07-05 10:32:04 IST!
```

Note After the system reboots, the self-signed and CA certificates of the servers, whose certificate type has changed, must be regenerated and re-uploaded into the client servers.