



Transactions

- [Transaction Logging and Audit, page 1](#)
- [View a Transaction, page 2](#)
- [Transaction Behavior, page 3](#)

Transaction Logging and Audit

Activity on the system results in transactions that are recorded. The Transaction menu provides auditing information for each transaction.

The recorded information includes:

- Transaction ID - identifier of the transaction
- Action - The type of action recorded in the transaction, for instance Execute, Create, Modify, Data Import and so on.
- Resource - the affected resource of the transaction, including the model type (for example, data/User) and its hierarchy.
- Username - Of the user who initiated the transaction.
- Submitted Time, Started Time and Completed Time - The date and time of the progress of the transaction.
- Rolled Back - Indicates whether the transaction was rolled back or not.
- Status - for running transactions, this is In Progress; for completed transactions it is Fail or Success.
- Detail - A brief description of the processed transaction.
- Duration - The duration of the selected transaction. If there are sub-transactions, this parent transaction duration is the total duration of the transaction.

On the GUI, details of a specific transaction are displayed when the transaction is selected from the list view.

When a transaction is selected, the **Base** tab shows details of the columns of the transaction list view. The button bar on the detail list view shows **Help** and **Refresh** buttons if the transaction is still running. If the transaction is running, click the **Refresh** button to update the Progress field.

If you want to cancel a transaction while it is still running, click the **Cancel** button. If a transaction, with sub-transactions, is cancelled, the sub-transaction currently in progress will complete. This sub-transaction

as well as all preceding sub-transactions will then roll back to their previous states. Note that bulk load transactions do not follow this behavior. Each bulk load sub-transaction is seen as a main transaction, and only the 'in progress' sub-transaction will roll back to its previous state.

A **Replay** button is available if the transaction is complete. A transaction can be replayed if required, for example if a transaction failed because a target system service was not running. The replay of the transaction can then be used instead of re-entering data on a GUI form.

An **Edit and Replay** button is also available for completed transactions. This is similar to the Replay button, but allows you to first make changes to the previously submitted form before the transaction is resubmitted.

The button is available for transactions that did not originate from bulk loads, wizards, or pop-up forms.

Replay and Edit and Replay functionalities are not supported by the bulk loader, because the bulk load files are not stored by default. The bulk loader extracts data from the spreadsheets and then performs the necessary action(s). The only time a bulk load file is stored in the database is when the bulk load is scheduled. In this case, the bulk loader keeps the file until it is triggered by the scheduler to execute the actions in the file. When the data is extracted from the file, it is deleted.

Selecting the button opens the original input form that resulted in the transaction. The form also contains the original data that was posted. This data can be edited and the form can be submitted to replay the transaction. This functionality can therefore be used, for example, to edit a failed transaction or to modify data of a successful transaction.

Since GUI Rules apply to a form from a specific hierarchy, the Edit and Replay functionality should only be used from the same hierarchy as the original transaction was executed.

If a transaction has sub-transactions, a Sub Transaction list is available of the form with links to their details. The sub-transaction form displays a link to a parent transaction.

Failed transactions show a Message of the error.

The Logs section on the Transaction base tab displays Message and Severity details of transactions performed by Cisco Unified Communications Domain Manager 10.6(1). For example, if the Severity has the status of error, the Message section can be expanded to inspect the error, and optionally copy it and send it to Support. If a workflow is inspected, a separate log entry provides details of each step with a log message as *Step n*, starting with Step 0.

The **Resource** tab, which has content for transaction types where a resource changed, displays the additional information, depending on the transaction type:

- Hierarchy - The point in the hierarchy at which the transaction occurred.
- Model Type - For example data/User.
- Current State - if available, click the Entity link to inspect the instance on the GUI form.

The **Back** button on the button bar can be used to navigate to the previous screen; for example, from the parent transaction screen to the list view of all transactions.

View a Transaction

You can only view transactions that are relevant to your specific hierarchy level. For instance, if you are logged into the system as a Customer Administrator you will be able to view all transactions that were performed at the customer for which you are the administrator. This includes transactions that were performed at any of the sites that belong to the customer. If you are logged in as a Site Administrator you will be able to view only the transactions that were performed at your specific site. The steps below can be followed on the GUI.

Procedure

-
- Step 1** Click **Administration Tools > Transactions**. The Transaction list view is displayed. The Transaction list view shows transactions in progress or executed. This is indicated in the Status column of the list. For completed transactions, the Success column indicates if the transaction was successful. The Description column provides additional details on the transaction if available.
- Step 2** Click an individual transaction (if required) to show a detailed view of the transaction.
- Step 3** If there are sub-transactions, click on the Link in the Sub Transactions list to show its details.
-

Transaction Behavior

The Cisco Unified Communications Domain Manager 10.6(1) transaction engine ensures that configuration changes are made efficiently and reliably.

In the event of a transaction failure or error, Cisco Unified Communications Domain Manager 10.6(1) allows for transactions to be rolled back to a state preceding the failed transaction.

For example, where a workflow step fails, all successful steps prior to a failed step are rolled back.

Transactions are hierarchical and have parent-child relationships with other transactions. Sub-transactions are always executed sequentially and synchronously, in other words the child transactions of a workflow parent transaction are executed one after another.

Transaction behavior is different for the following actions in the system:

- API

The API supports executing transactions in both synchronous and asynchronous modes. When executed in synchronous mode the API responds only once the transaction has completed. When executed asynchronously, the API responds immediately with a transaction ID so that the progress and status of the transaction can be polled.

- Bulk Loaders

With bulk loading, the load of each row on a sheet is a separate transaction. These transactions are run in series. There is no rollback of rows that have loaded successfully prior to, or subsequent to, a failed transaction (a failed row on a sheet). Multiple bulk load sheets can be loaded in parallel.

- Data Import

A single transaction is created for each record in the import file. If a single transaction fails, the import continues and does not roll back the preceding successful transactions.

- Data Sync

A single parent transaction is created for a data sync action. The subsequent device API requests are not handled as sub transactions but are executed in-line.

- Events

Events can be triggered as part of data sync operations or as triggers on operations performed on certain model types. The provisioning workflow executed when the event triggers is executed as a new parent

transaction. Transaction failures with the workflow executed after an event do not affect the original transaction that triggered the event.

All transactions are placed on a queue before they are actioned. Parent transactions can run concurrently, but their subtransactions run serially. There is priority in parent transactions so that user input such as adding on a GUI form will be prioritized over a running import or bulk load process.