CA Service Catalog

Reference Guide

Release 12.8.00

technologies

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Technologies Product References

This documentation set contains references to the following CA Technologies products:

- CA Service Catalog, which includes Accounting Component (formerly CA Service Accounting)
- CA Embedded Entitlements Manager (CA EEM)
- CA Server Automation
- CA Server Automation Reservation Manager (Reservation Manager)
- CA Business Service Insight (CA BSI, formerly CA Oblicore Guarantee)
- CA Open Space
- CA Role & Compliance Manager (CA RCM)
- CA Service Desk Manager, which includes CA CMDB
- CA SiteMinder®
- CA Asset Portfolio Management (CA APM)
- CA MICS® Resource Management
- CA JARS®
- CA Storage Resource Manager (CA SRM)
- CA Workflow
- CA Process Automation (formerly CA IT PAM)
- CA Business Intelligence
- CA Anti-Virus (formerly eTrust Antivirus)
- CA Threat Manager (formerly eTrust Integrated Threat Management [eTrust ITM])

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Utilities

CA Service Catalog provides several command line utilities.

This section contains the following topics:

IXUTIL Import/Export Utility

IXUTIL is an importing and exporting command line utility that allows the preservation and migration of CA Service Catalog data between systems. For example, you can export offerings (services) from a CA Service Catalog Release 12.7 system and import them as services into a CA Service Catalog Release 12.8 system, using the same XML structure.

You can use IXUTIL to export and import new and updated objects from one CA Service Catalog instance to another CA Service Catalog instance. This migration is primarily intended for both of the following:

- Moving data from a test computer to a production computer, when both computers are running the same release of CA Service Catalog (same-version migration)
- Moving data from a computer running the previous release of CA Service Catalog to a computer running a newer release of CA Service Catalog (upgrading)

IXUTIL does *not* import or export *images* of offerings because of the variation of the image file path names on the source and target computers. If you want to retain such images, you must copy and paste them manually, after the initial migration is complete.

Note: Use the CA Workflow IDE, *not* IXUTIL, to export and import CA Workflow actors and process definitions between systems running the *same* release of CA Service Catalog; for details, see the CA Workflow IDE online help.

IXUTIL Files

For Windows, the import/export utility file (ixutil.bat) is located in the USM_HOME\scripts folder.

You can configure IXUTIL by using the ixutil.cfg properties file located in the USM_HOME\scripts folder. This file controls import and export rules, default export file format, error log file and so on. The ixutil.cfg file contains comments and details about each configurable parameter.

Log messages from IXUTIL appear in the USM_HOME\logs\install\ixutil.log file.

Data File Types

For most object types, IXUTIL exports the data in a proprietary XML format, and you must use the same XML format when importing. You can determine the correct XML formats for each object type by examining an exported XML file.

IXUTIL lets you import and export the following types of data:

- Catalog Definitions
  - Services
    - With the definition
    - With associated objects

      Examples include Service Level Agreement information, CMDB CI Mapping, permissions, business hours, resource type, application metric, policies and actions, and translations.
- Reporting components
  - Data objects
- Administrative components
  - Events, rules, and actions
  - Forms
  - Policies
  - Configurations
  - Business units

IXUTIL Export Syntax

The export syntax for using the ixutil.bat (Windows) command is as follows:

ixutil export *object* –f *file* –format xml *object-specific parameters*

object

Specifies one of the data types.

-f *file*

Specifies the name of the file to be exported to; if omitted, the data is exported to the command window, when the name of the file is provided then the file is created in the default directory that is specified in ixutil.cfg file. The Out of the Box folder is USM_HOME\view\exports in ixutil.cfg file.

-format xml

Indicates the format of the exported data, when omitted, the default is taken from the ixutil.cfg file; all objects support XML output only.

object-specific parameters

Indicates that additional parameters can be used depending on the object value.

IXUTIL Import Syntax

The import syntax for using the ixutil.bat (Windows) command is as follows:

ixutil import *object* –f *file* –format xml *object-specific parameters*

object

Specifies one of the objects, using the object syntax (see page 12)

-f *file*

Specifies the name of the file to be imported from. If no folder path is provided, only a file name, the folder used is USM_HOME\scripts.

USM_HOME specifies the CA Service Catalog installation directory on this computer. For 32-bit computers, the default path name is C:\Program Files\CA\Service Catalog. For 64-bit computers, the default path name is C:\Program Files (x86)\CA\Service Catalog.

-format xml

Indicates the format of the file to be imported, if omitted, the default is taken from the ixutil.cfg file; most objects support XML input only.

object-specific parameters

Indicates that there may be additional parameters that can be used depending on the object value.

Object Syntax

In most cases, you must import or export each type of object in its own separate command, as shown in the following list. For example, you import services, and events in three separate commands. You cannot combine two or more types of objects in a single import command. An exception is the service offering option that contains service option group and service option element as noted in the following list.

Note: You do not literally enter the brackets around these options; the brackets are in this list for readability only.

Configurations

```
ixutil import configuration [-f file] [-format xml] [-businessunit id] [-group name] [-contentpack name]
```

```
ixutil export configuration [-f file] [-format xml] [-businessunit id] [-group name]
```

Events

```
ixutil import event [-f file] [-format xml] [-event name] [-rule name] [-action name] [-contentpack name] [-disable] [-disallow_duplicates]
```

```
ixutil export event [-f file] [-format xml] [-event name] [-rule name] [-action name]
```

Forms

```
ixutil import form [-f file] [-format xml] [-cawf -businessunit id] [-contentpack name] [-modified_only] [-disallow_duplicates]
```

```
ixutil export form [-f file] [-format xml] [-folder foldername][-form formname] [-businessunit id]
```

Services

```
ixutil import service [-f file] [-businessunit id] [-include_incomplete_sub] [-include_cmdb_ci_mapping] [-include_permissions] [-include_request_sla] [-inlcude_service_hour] [-include_resource_type] [-include_application_metric] [-contentpack name] [-modified_only] [-include_associated_policies_and_actions] [-disallow_duplicates] [-include_translation] [-merge_translation]
```

```
ixutil export service [-f file] [-folder name] [-servicename name] [-businessunit id] [-include_forms] [-include_cmdb_ci_mapping] [-include_permissions] [-include_request_sla] [-include_service_hour] [-include_resource_type] [-include_application_metric] [-include_associated_policies_and_actions] [-include_translation] [-extract_translation]
```

Policies

```
ixutil import policy [-f file] [-format xml] [-businessunit id] [-disable] [-contentpack name]
```

```
ixutil export policy [-f file] [-format xml] [folder foldername[,foldername2]] [-policy policyname[,policyname2]]
[-businessunit id]
```

Report Data

```
ixutil import report_data [-f file] [-format xml] [-folder name] [-contentpack name] [-disallow_duplicates]
```

```
ixutil export report_data [-f file] [-folder name(s)] [-id(s)]
```

Business Units

```
ixutil import businessunit [-f file] [-format xml] [-businessunit id]
```

```
ixutil export businessunit [-f file] [-businessunit id]
```

Object Syntax Rules and Considerations

The following rules apply to the IXUTIL command line:

- Businessunit id refers to the Business Unit Login ID.
- Parameter values that contain spaces must be in double quotes.
- For lists, separate values with commas.
- Date values must use the format YYYY/MM/DD H24:MM:SS.
- The Report-related object, report_data must be imported to support interdependency.

The following considerations apply to the IXUTIL command line:

- The “tenant” and “domain” references are to old terminology from prior releases. The current terminology for “tenant” and “domain” is “business unit”. For example, to export all the sub-business units for the “CA” business unit, use:

  ```
  ixutil export businessunit –f output.xml –businessunit “CA”
  ```

- The “offering” and “rateplan” references are to old terminology from prior releases. The current terminology for “offeringrateplan” is “service” and for “rateplan” is “service option group”. For example, to export all the services for the “CA” business unit, use:

  ```
  ixutil export service –f output.xml -businessunit “CA”
  ```

IXUTIL Parameters

The following parameters are used for the IXUTIL command:

-businessunit

Specifies the Business Unit. For exporting, when the domain (businessunit) parameter is used, the business unit that is specified must exist in the source system to export the objects of that business unit.

When specified during import, business unit refers to the business unit in which the objects are imported.

For exporting policy files, this parameter specifies the business unit that contains the policies being exported. If you are exporting all policies in all business units, omit this parameter; otherwise, this parameter is required.

For importing policy files, you can optionally specify this parameter to import all the policies into a single business unit, as follows:

- Use the -businessunit option to specify a business unit when you import the file on the target computer. The ixutil command writes all the policies in the file to the business unit you specify.

If that business unit does not exist, the import fails and displays an error message similar to the following:

```
Invalid Login Domain specified
```

- Do *not* specify the -businessunit option when you import the file on the target computer. The ixutil command writes all the policies in the file to the business unit specified in the file.

If that business unit does not exist, the policies assigned to it are instead assigned to the root (top-level) business unit.

-event eventname

Specifies the name of the event as seen in the Event Type list (display_name field in the usm_rule_event_type table). This option is used to export and import event, rule, and action as it appears on the Administration, Tools, Events-Rules-Actions page.

If the name contains spaces, enclose the name in double quotation marks, as follows:

```
ixutil .. -event "Account Change"
```

To specify multiple names, enclose the names in double quotation marks, and separate the names with commas, as follows.

```
ixutil .. -event "Request/Subscription Item Change" -rule "When Status is Pending Approval,When Status is Pending Fulfillment" -action "Notify Requestor on Pending Approval,Launch USM_FulfillmentUSDRequest Workflow"
```

-rule rulename

Specifies the name of the rule as seen in the Rule Type list. This option is used to export and import a rule information about the specified event type.

-action actionname

Specifies the name of the action as seen in the Action Type list. This option is used to export and import action information about the specified event type.

Note: If you specify the -rule option, you must specify the -event option, and the event must contain the rule. Similarly, if you specify the -action option, you must specify the -event and -rule options, and the rule must contain the action.

-file

Specifies the name of the file being exported or imported. This file may contain the policy object or objects that you are exporting or importing. The Catalog system uses an XML file to export and import policy objects.

-folder

Specifies the folder from which the objects are exported or the folder into which the objects are imported.

Important! When you run an import command, an error occurs if you specify the name of a folder that does not exist: The system does *not* create the folder and does *not* import the objects.

-format

Specifies the format of the export file. By default XML is the format that is used if the *–format* parameter is omitted. Currently, the only supported export format is XML.

-include_incomplete_sub

Imports and then deletes the objects (such as services and service option groups) that belong to a subscription that has not yet completed. Use this option when importing offerings (services) or rate plans (service option groups).

If you do not specify this option, the objects that belong to such subscriptions are imported and retained (not deleted).

IXUTIL verifies for subscriptions in all services, including inherited services, child services, and inherited services of child services.

-include_cmdb_ci_mapping

Specifies that the associations between CA Service Catalog services and CA CMDB configuration items are exported when you use the ixutil -export command and imported when you use the ixutil -import command. The -include_cmdb_ci_mapping option applies *only* when you have integrated CA Service Catalog with CA CMDB. For details, see the *Integration Guide*.

Important! Verify that the CA CMDB configuration items are already present on the target computer *before* you run the ixutil command. Copy them to the target computer, if necessary. Also verify that the UUIDs of these configuration items are the same on both the source and target computers. For instructions to copy the configuration items and verify the UUIDs, see your CA CMDB documentation.

-include_request_sla

Exports or imports the associated request SLAs for each service (offering) while exporting or importing service through service commands.

Sample commands follow:

```
ixutil export service –f serviceSLAs.xml –servicename blackberry –include_request_sla
```

```
ixutil import service –f serviceSLAs.xml –include_request_sla
```

-include_service_hour

Exports or imports the associated service hours (outage calendars and business hours) for each service (offering) while exporting or importing services through service commands.

-group_name

Specifies the configurations that are grouped under various categories. Many groups are configured in the Administration Configurations, Accounting Configurations, Catalog Configurations, and Custom Configurations. The Configuration name to be provided as input in the Group Name text box is displayed in brackets beside the configuration name in the header section in the Graphical User Interface. You can import or export multiple groups at the same time. To export multiple groups, separate the names with commas.

For example, to export a CA CMDB configuration group, provide the input as CMDB. All the configurations under this group are exported or imported. Leave this field blank to export all the configurations irrespective of the group.

Note: The group_name must be in the format <businessunitid>_<cc>_<configuration_form_id> to export the custom configurations.

-include_associated_policies_and_actions

Exports or imports the associated policies and actions for each service (offering) while exporting or importing services through service offerings.

-include_permissions

Exports or Imports the permissions for each service (offering), service option group, and service option element associated with the service.

Important! Restart the CA Service Catalog Windows service to see the new permissions in the product UI.

-include_resource_type

Exports or Imports the resource type that is associated with the service (offering).

-include_application_metric

Exports or Imports the application, its metric, and the definition of the service that is associated with the Application Rate item.

-include_translation *and* -merge translation

Apply *only* when you localize multiple services. This process is documented in the *Localizing Multiple Services* scenario.

Note: For details, see this scenario in the *Administration Guide* or on http://ca.com/support.

-disallow_duplicates

Applies only to the import command of different objects. If there is a same object with a similar name and different tracking id (a different object with same name) while importing CA Service Catalog system, then that object is updated and no new object is created. If this parameter is not specified, then a new object with the same name is created. This parameter is important in two circumstances:

- While importing CA Service Catalog system that already has same named objects that are either created on that system or imported from a different system.
- When you do not want to create new objects during import but still want the already existing objects to be updated.

-modified_only

Specifies that only the modified objects are imported based on comparison of date last modified. IXUTIL imports all the objects by default without any date last modified comparisons. You must specify this parameter only when you want to import the modified objects. This parameter applies to the service and form objects only.

-disable

Specifies that the objects are imported in the disabled state. In event rule actions, the rules, actions are in disabled state and in the case of policy the objects are in inactive state.

By default, policies, events, rules, and actions are imported and exported in their current status. For example, if you export a policy whose status is disabled, the policy remains disabled when you import it. Similarly, if you export a policy whose status is enabled, the policy remains enabled when you import it. However, you can add the -disable option to the command to import the policy (or policies) in disabled status.

Note: The -disable option applies only to the import command; it does not apply to the export command.

Use the form-related parameters to export and import CA Workflow forms and Form Designer forms. The following are the included parameters:

- -form *formname* and -folder *foldername*

- -include_forms for Form Designer forms
- -cawf for CA Workflow forms

  You can use these parameters to perform the following tasks:

  - Import CA Workflow forms into the Form Designer (see page 28)
  - Import and export Form Designer forms (see page 29)
  - Import and export Form Designer forms associated to service option elements (see page 32)

Object Modification

An object can be modified in the following ways:

- Modifying the objects that it contains. For example, a service contains service option groups. A service option group, in turn, contains service option elements.
- Modifying the properties of an object, such as its date available, description and so forth.

Note: The consideration for object modifications and the conditions for date last modified are only required if the -modified_only parameter is specified. Refer to the description of -modified_only parameter in IXUTIL Parameters (see page 14).

Object Modification Considerations

Consider the following information when you use IXUTIL for services, service options, and service option groups.

The term *objects* here means services, service option groups, and service option elements. As a best practice, do *not* modify objects directly on your production computer. Instead, always modify objects on your test computer first, test them thoroughly, and then export them to the production computer. If you do update objects on both the test and production computers and then export, you may accidentally overwrite changes or data on the production computer. You *cannot* merge changes between two same-named objects on the source and destination computers. The newer object always take precedence when you run IXUTIL.

The following considerations apply to services, service options, and service option groups objects:

- Two services are assumed to be same if they have the same name and belong to same business unit.
- Two service option groups are assumed to be the same if they have the same name and belong to same business unit.
- In the tables named usm_offering and usm_rate_plan, the -date_last_modified field is used to determine if an object has been modified since the last import. For example, to find all services that have been modified since a specific date, for example, January 1, 2009, you can run a query in the Report Builder on the usm_offering table.

Date Last Modified Conditions

The date last modified is updated under the following conditions:

- Creating, editing, or deleting a service option element updates the time stamp of any service option group or service to which it belongs, including all parent services.
- Creating, editing, or deleting a service option group updates the time stamp of the service option group itself and any service to which it belongs, including all parent services.
- Creating, editing, or deleting a service updates the time stamp of the service itself and of any parent services in the same path. Editing a service means that a service option group is either added to or deleted from the service, or that the properties such as date available, date unavailable, and so forth are modified.
- Cutting and pasting (moving) a service from one folder to another updates the time stamp of the service, the folder to which it was moved (the new parent folder), and any parent services in the same path. The moved object uses the ID and the properties of the new parent folder, such as parent_id and path name.

  Note: The time stamp of the folder from which the service was moved (the old parent folder) is *not* updated, for these reasons: no attributes of the old parent folder are changed, and no attributes of the moved service use the old parent folder.
- A new inherited service receives a *new* time stamp, *not* the time stamp of the service on which it depends. Also, the time stamp of the new inherited service is updated when any of the following events occur. In the following list, the service means the service under which the new inherited service was placed.
  - A service option group is added to or removed from the service
  - A child service is moved from one location to another
  - A child service is inherited

Export and Import All Objects

You can export all objects to the target computer from an existing CA Service Catalog implementation. All the objects in the XML file are imported without any timestamps checks. Use the –modified_only parameter to view the previous behavior of IXUTIL. This parameter is basically introduced for backward compatibility. You can export all objects in the following ways:

- Export and import offerings and *associated* service option groups simultaneously in a single command by using the service option, by entering the following commands on the source and target computers, respectively:

  ```
  ixutil export service -f allservices.xml
  ```

  ```
  ixutil import service -f "pathname\allservices.xml"
  ```

Preservation of Inheritance

When you export services that contain service option groups, and service option elements from one business unit to another, inheritance is supported if you follow these rules. If you do not follow these rules, you can export the objects, but the inheritance is broken.

- Export the objects within the same business unit.
- Export the objects as a single xml file.

  For example, Service A and Service B have a parent-child relationship in the business unit named BU-x. Use following commands *in sequence* to export the services with inheritance preserved:

  ```
  ixutil export service -f output.xml -servicename A,B –businessunit BU-x
  ```

  ```
  ixutil import service -f output.xml –businessunit BU-y
  ```

- The parent object *must* exist in the target system before you import the file.

  Continuing the previous example, enter the following commands *in sequence* to export the services with inheritance preserved:

  ```
  ixutil export service -f output1.xml -servicename A –businessunit BU-x
  ```

  ```
  ixutil export service -f output2.xml -servicename B –businessunit BU-x
  ```

  ```
  ixutil import service -f output1.xml –businessunit BU-y
  ```

  ```
  ixutil import service -f output2.xml –businessunit BU-y
  ```

Export and Import Events, Rules, and Actions

You can use IXUTIL to export and import new and updated objects, including events, rules, and actions, from one computer to another. You can export and import services in the following cases:

- When you upgrade or migrate CA Service Catalog
- When you upgrade your computer
- When the server fails

Note: You can also use the Import Export Utility on the product UI to import and export configurations. The Import Export Utility provides a GUI to import and export configurations and other objects that is not present for Events.

Follow these steps:

1. Open the CA Service Catalog command prompt on any Catalog Component computer.
2. Navigate to the Scripts folder in the command prompt.
3. Enter one or more of the following export commands on the source computer, as applicable:
   - To export all events enter the following command. The command exports all rules and actions in each event.

     ixutil export event -f *file* -format xml

   - To export an event, use the following command. The command exports the specified event, including all its rules and actions.

     ixutil export event -event *eventname* -f *file* -format xml

   - To export an event with only the specified rule, use the following command. The command exports the specified event, with the specified rule and all its actions.

     ixutil export event -event *eventname* -rule *rulename* -f *file* -format xml

   - To export an event with only the specified rule and the specified action, use the following command. The command exports the specified event, with the specified rule and the specified action.

     ixutil export event -event *eventname* -rule *rulename* -action *actionname* -f file -format xml

   Refer to IXUTIL Parameters (see page 14) for more information on the parameters.

4. Enter one or more of the following import commands on the target computer, as applicable:
   - To import all events, enter the following command. The command imports all rules and actions in each event.

     ixutil import event -f *file*

- To import an event, use the following command. Optionally add the -domain parameter from the previous bullet to specify a business unit. The command imports the specified event, including all its rules and actions.

  ```
  ixutil import event -event eventname -f file
  ```

- To import an event with only the specified rule, use the following command. The command imports the specified event, with the specified rule and all its actions.

  ```
  ixutil import event -event eventname -rule rulename -f file
  ```

- To import an event with only the specified rule and the specified action, use the following command. The command imports the specified event, with the specified rule and the specified action.

  ```
  ixutil import event -event eventname -rule rulename -action actionname -f file
  ```

- To import an event (including all its rules and actions) in disabled status, add the -disable option, as shown in the following example.

  ```
  ixutil import event -event eventname -f file -disable
  ```

- To import one rule and its action in disabled status, add the -disable option, as shown in the following example.

  ```
  ixutil import event -event eventname -rule rulename -action actionname -f file -disable
  ```

5. If you *imported* events, rules, or actions, restart the Windows service named CA Catalog Component. Restart this service on *all* Catalog Component computers in your environment.

   Important! If you import objects but do not restart the CA Catalog Component service on *all* Catalog Component computers, unpredictable results can occur.

6. Log in to CA Service Catalog on the computer where you imported the objects.
7. Click Administration, Tools, Events-Rules-Actions.

   The Event Types page appears.

8. View the events, including the rules and actions. Verify that the Catalog system imported the objects as you wanted.
9. Close the CA Service Catalog command prompt.

You have exported and imported events, rules, and actions.

Export and Import Configurations

You can use IXUTIL to export and import new and updated objects, including configuration objects, from one computer to another. You can export and import configurations in the following cases:

- When you upgrade or migrate CA Service Catalog
- When you upgrade your computer
- When the server fails

Note: You can also use the Import Export Utility on the product UI to import and export configurations. This utility provides a GUI for importing and exporting configurations and other objects.

Follow these steps:

1. Open the CA Service Catalog command prompt on any Catalog Component computer.
2. Navigate to the Scripts folder in the command prompt.
3. Enter one or more of the following commands:
   - To export all the configurations in all the business units, enter the following command:

     ```
     ixutil export configuration -f file
     ```

   - To export all the configurations in a business unit, enter the following command:

     ```
     ixutil export configuration -f file - businessunit businessunit id
     ```

   - To import the groups in a business unit, enter the following command:

     ```
     ixutil import configuration -f file -format xml -businessunit businessunit id -group name
     ```

     Note: You must restart the View service after importing the configuration.

     Refer to IXUTIL Parameters (see page 14) for more information on the parameters.
4. Log in to CA Service Catalog on the computer where you imported the configurations. If you had imported configurations into a specific business unit, log in to the respective business unit.
5. Click Catalog, Configuration, or Administration, Configuration, or Accounting, Configuration.
6. Verify that CA Service Catalog imported the configurations as you want.
7. Close the CA Service Catalog command prompt.

Export and Import Services

You can use IXUTIL to export and import new and updated objects, including service objects, from one computer to another. You can export and import services in the following cases:

- When you upgrade or migrate CA Service Catalog
- When you upgrade your computer
- When the server fails

Note: You can also use the Import Export Utility on the product UI to import and export services. This utility provides a GUI for importing and exporting services and other objects.

Follow these steps:

1. Open the CA Service Catalog command prompt on any Catalog Component computer.
2. Navigate to the Scripts folder in the command prompt.
3. Enter one or more of the following commands:
   - To export all services in all business units, enter the following command:

     ```
     ixutil export service -f file
     ```

   - To export all services in a business unit, enter the following command:

     ```
     ixutil export service -f file - businessunit businessunit id
     ```

   - To export a folder containing one or more services, enter the following command:

     ```
     ixutil export service -f file -folder name -businessunit id
     ```

     To export multiple folders, enclose the names in double quotation marks, and separate the names with commas.

   - To export a folder containing offerings with the cmbd ci mapping, permissions, forms, resource types, sla, service hours, associated policies and actions, and transactions enter the following command:

     ```
     ixutil export service -f file -folder name -servicename name -businessunit id -include_forms
     -include_cmdb_ci_mapping -include_permissions -include_request_sla -include_service_hour
     -include_resource_type -include_application_metric -include_associated_policies_and_actions
     -include_translation -extract_translaton
     ```

   - To import a previously exported file containing one or more services, enter the following command:

     ```
     ixutil import service -f file -businessunit id
     ```

     The -businessunit *business unit id* is optional when you import the service. However, consider the important factors explained in the description of this option later in this topic.

- To import a previously exported file that includes the incomplete sub, cmbd ci mapping, permissions, associated policies and actions, and transactions, enter the following command:

  ```
  ixutil import service -f file -businessunit id -include_incomplete_sub -include_cmdb_ci_mapping
  -include_permissions -include_request_sla -inlcude_service_hour -include_resource_type
  -include_application_metric -contentpack name -modified_only
  -include_associated_policies_and_actions -disallow_duplicates -include_translation -merge_translation
  ```

  Refer to IXUTIL Parameters (see page 14) for more information on the parameters.

4. Log in to CA Service Catalog on the computer where you imported the services. If you had imported services into a specific business unit, log in to the respective business unit..

5. Click Catalog, Service Offerings.

6. Verify the following:

   - Imported folders or offerings

   - Imported Associations

7. Close the CA Service Catalog command prompt.

Export and Import Report Data

You can use IXUTIL to export and import new and updated objects, including report data, from one computer to another. You can export and import report data in the following cases:

- When you upgrade or migrate CA Service Catalog
- When you upgrade your computer
- When the server fails

Note: You can also use the Import Export Utility on the product UI to import and export report data. This utility provides a GUI for importing and exporting report data and other objects.

Follow these steps:

1. Open the CA Service Catalog command prompt on any Catalog Component computer.
2. Navigate to the Scripts folder in the command prompt.
3. Enter one or more of the following commands:
   - To export all report data enter the following command:

     ```
     ixutil export report_data -f file
     ```

   - To export a folder containing one or more report data, enter the following command:

     ```
     ixutil export report_data -f file -folder name(s) -id name(s)
     ```

     To export multiple folders, enclose the names in double quotation marks, and separate the names with commas.

   - To import a folder containing one or more report data that includes content pack, enter the following command:

     ```
     ixutil import report_data -f file -format xml -folder name -contentpack name -disallow_duplicates
     ```

   Refer to IXUTIL Parameters (see page 14) for more information on the parameters.
4. Log in to CA Service Catalog on the computer where you imported the report data. If you had imported report data into a specific business unit, log in to the respective business unit.
5. Click Administration, Report Builder, Data Objects.
6. Verify that the CA Service Catalog imported the Report Data as you want.
7. Close the CA Service Catalog command prompt.

Export and Import Business Units

You can use IXUTIL to export and import new and updated objects, including business unit objects, from one computer to another. You can export and import business units in the following cases:

- When you upgrade or migrate CA Service Catalog
- When you upgrade your computer
- When the server fails

Note: You can also use the Import Export Utility on the product UI to import and export business units. This utility provides a GUI for importing and exporting business units and other objects.

Follow these steps:

1. Open the CA Service Catalog command prompt on any Catalog Component computer.
2. Navigate to the Scripts folder in the command prompt.
3. Enter one or more of the following commands:
   - To export all business units enter the following command:

     ```
     ixutil export businessunit -f file -businessunit id
     ```

     Where businessunit id is the root BU (businessunit) id.
   - To import all business units enter the following command:

     ```
     ixutil import businessunit -f file -format xml -businessunit id
     ```

   Refer to IXUTIL Parameters (see page 14) for more information on the parameters.
4. Log in to CA Service Catalog on the computer where you imported the business units. If you had imported business units into a specific business unit, log in to the respective business unit.
5. Click Administration, Business Units.
6. Verify that the CA Service Catalog imported the Business Units as you want.
7. Close the CA Service Catalog command prompt.

Import CA Workflow Forms into the Form Designer

You can optionally import CA Workflow forms into the Form Designer. For example, you can do so if you upgraded from an earlier release of CA Service Catalog and have existing CA Workflow forms that you want to convert to Form Designer forms before continuing to use them in your services. Converting CA Workflow forms to Form Designer forms can be especially helpful as part of changing your process automation tool from CA Workflow to CA Process Automation.

Follow these steps:

1. In CA Workflow, select the form or forms that you want to import, and verify their path names.
2. Verify that the forms are already present on the target computer; copy them manually to the target computer, if necessary.

   CA Workflow forms are stored as xml files. If you are using a filestore, they are stored in the FileStore\forms folder.
3. On any Catalog Component computer, open a command prompt and enter the following command:

   ```
   ixutil import form -f {file|foldername} -format xml -cawf -businessunit id
   ```

   -cawf

   Imports CA Workflow forms and converts them to Form Designer forms. The CA Workflow forms can be either pre-defined or customized.

   Note: The -cawf option is required to import Form Designer forms.

   -businessunit id

   Specifies the business unit login ID, the name used to log in to the business unit.

   For example, if the business unit login ID is subBud, specify –businessunit subBud.

   When you use the –businessunit *businessunit-id* option, the imported objects overwrite any existing objects in the domain that have the same name. For example, if you import a service (offering) named Diamonds using the option "-businessunit ABC," then the imported service named Diamonds overwrites the existing service named Diamonds (if one exists) in the businessunit named ABC.

   -f ***file***

   To import an individual CA Workflow form, specify -f *file*, where *file* specifies the complete path name of the form.
4. Verify that you can view and edit each imported form in the Form Designer.

5. If necessary, make any formatting changes or any other required updates to the form.
6. When the newly imported Form Designer form meets your requirements, update the service option to remove the CA Workflow form and to include the Form Designer form.

Export and Import Form Designer Forms

You can optionally export and import Form Designer forms alone, *without* any related service option elements, from a source (test) computer to a target (production) computer. Do so when you want to move and re-use one or more forms but not the related service option elements. You can export and import any of the following:

- All forms in a domain (business unit)
- All forms in all business units
- All forms in a folder
- A specific form.

Follow these steps:

1. On the source Catalog Component computer, export the form or forms by opening a command prompt and entering the following command:
2. Navigate to the Scripts folder in the command prompt.
3. Enter one or more of the following commands:

```
ixutil export form -f file -format xml {-folder foldername|-form formname} -businessunit id
```

-f *file*

Specifies the complete path name of the XML file being exported; specify an intuitive file name. This file contains the forms, service option elements, and other service-related entities that you want to move to the target computer.

-businessunit *id*

Specifies the business unit login ID, the name used to log in to the business unit. Use this option to export all forms in a domain. If you omit this option, the command exports all forms in all businessunits.

For example, if the business unit login ID is subBud, specify –businessunit subBud.

{-form *formname*|-folder *foldername*}

Exports either an individual Form Designer form or all Form Designer forms from a specified folder, as follows:

To export an individual Form Designer form, specify -form *formname*, where *formname* specifies the name of the form being exported.

To export all the Form Designer forms from a specific folder, use -folder *foldername*, where *foldername* specifies the name of the folder being exported.

Note: For Form Designer forms and folders, use only the form name or folder name; path name is *not* applicable.

The command runs and creates the XML export file at the location USM_HOME/build/install/view/exports. The file name is the name you specify using the -f *file* parameter described earlier.

Sample commands follow:

```
ixutil export form -f file1.xml -format xml -form "Sample Form" -businessunit subBud
```

```
ixutil export form -f file2.xml -format xml -folder "My Forms" -businessunit subBud
```

4. Copy the exported XML file to the target Catalog Component computer.

5. On the target Catalog Component computer, open a command prompt and enter the following command to import the XML file that you exported and copied in the previous steps:

```
ixutil import form -f file -format xml -businessunit id
```

-f *file*

Specifies the complete path name of the XML file being imported.

Note: You already specified form, folder, and domain information when you ran the export command that created this XML file (see Step 1); therefore, in the import command, you specify *only* this XML file name.

-businessunit *id*

Specifies the business unit login ID, the name used to log in to the business unit. The exported XML file is imported into this business unit.

When you use the –businessunit *id* option, the imported objects overwrite any existing objects in the domain that have the same name. For example, if you import a service (offering) named Diamonds using the option "-businessunit ABC," then the imported service named Diamonds overwrites the existing service named Diamonds (if one exists) in the businessunit named ABC.

The command runs and imports the form or forms and folders (if applicable) into the target (production or backup) computer, as follows:

- If a form or folder being imported does *not* already exist on the target computer, it is imported normally and becomes available.

Note: The timestamp or the date last modified is no longer available. All that is part of XML file is imported. If you want the timestamp check then you should use the –modifed_only parameter.

If you import a folder, then that folder, its forms, and its child folders are imported, with their folder structure maintained. However, *both* that folder *and* its parent folder from the source computer are attached to the root folder (named Forms) on the target computer. If the parent folder contains other child folders besides the one you exported, those other child folders are also exported and are included under the parent folder, with their folder structure maintained.

For example, on the source computer, suppose Folder1 contains Folder2, and Folder2 contains Form1 and Form2. If you export folder 2 from the source computer and import it on target computer, then both Folder1 and Folder2 become root level folders on the target computer. Form1 and Form2 remain under Folder 2. Folder1 on the root folder includes and any forms or child folders that belong to it (except for Folder2).

6. Verify that the forms appear correctly in the Form Designer; update them if necessary.
7. Test the newly imported forms.
8. Make the forms available for the users.

Export and Import Form Designer Forms Associated to Service Option Elements

You can export and import Form Designer forms to include them when exporting and importing service that contains service option elements (rate items), for example, when moving from source (test) to target (production) or from backup to restore.

In IXUTIL export commands for service that has service option elements, specify the –include_forms option to include the Form Designer forms (not CA Workflow forms) associated with the service option elements. The –include_forms option is not valid in IXUTIL import commands; the forms are included in the import automatically if they were included in the export.

Follow these steps:

1. On the source Catalog Component computer, enter the following commands to export the folder structure for the forms from the source computer and import it on the target computer. If you do not want to export and import the folder structure, skip this step.

   ixutil export form -f *file* -format xml {-folder *name*|-form *name*} *-businessunit id*

   ixutil import form -f *file* -format xml *-businessunit id*

   For an explanation of the command options, see the descriptions of these commands in Import and Export Form Designer Forms (see page 29).

2. On the source Catalog Component computer, export the forms associated with the service option elements by opening a command prompt and entering the following command:

   ixutil export {service} -f *file* {*-offering name*|*-folder name*|*-businessunit id*} -include_forms *other options*

   service

   Includes services (offerings), service option groups (rate plans), and service option elements (rate items).

   Note: Service option elements are processed in IXUTIL export service commands *only*. Therefore, use –include_forms in IXUTIL export service commands *only*, as explained in this topic.

   -include_forms

   Includes Form Designer forms (not CA Workflow forms) associated with service option elements (rate items).

   {-offering *name*|-folder *name*|*-businessunit id*}

   Restricts the forms being exported to those that belong only to a specific service (-offering *name*), folder (-folder *name*), or *businessunit* (*-businessunit id*).

   The command runs and creates the XML export file at the location USM_HOME/build/install/view/exports. The file name is the name you specify using the -f *file* parameter.

3. Copy the exported XML file to the target Catalog Component computer.

4. On the target Catalog Component computer, open a command prompt and enter the following command to import the forms:

   ```
   ixutil import {service} -f file other options
   ```

   If you specified *service* in the original export command, specify *service* in the corresponding import command.

   -f file

   Specifies the complete path name of the XML file being imported.

5. Verify that the forms appear correctly in the Form Designer; update them if necessary.

6. Test the newly imported forms.

7. Make the forms available for the users.

Export and Import Policies

The IXUTIL command line utility is an importing and exporting command line utility that allows the preservation and migration of CA Service Catalog data between computers. You can use IXUTIL to export and import new and updated objects, including policies, from one computer to another. You can export and import policies in the following cases:

- When you upgrade or migrate CA Service Catalog
- When you upgrade your computer
- When the server fails

Note: You can also use the Import Export Utility on the product UI to import and export policies. This utility provides a GUI for importing and exporting policies and other objects.

Follow these steps:

1. Open the CA Service Catalog command prompt on any Catalog Component computer.

2. Enter one or more of the following commands:

   - To export all policies in all business units, enter the following command:

     ```
     ixutil export policy -f file
     ```

   - To export all policies in a business unit, enter the following command:

     ```
     ixutil export policy -f file - businessunit businessunit id
     ```

- To export a single policy, use the following command:

  ```
  ixutil export policy -f file -policy name -businessunit id
  ```

  If the name contains spaces, enclose the name in double quotation marks, as follows:

  ```
  ixutil export policy -policy "Mobile Device Policy--Tier 1" ...
  ```

- To export multiple policies, enclose the names in double quotation marks, and separate the names with commas, as follows.

  ```
  ixutil export policy -policy "Mobile Device Policy--Tier 1,Mobile Device Policy--Tier 2,Mobile Device Policy--Tier 3" ...
  ```

- To export a folder containing one or more policies, enter the following command:

  ```
  ixutil export policy -f file -folder name -businessunit id
  ```

  To export multiple folders, enclose the names in double quotation marks, and separate the names with commas. Use the previous command for multiple polices as a model.

- To import a previously exported file containing one or more policies, enter the following command:

  ```
  ixutil import policy -f file -businessunit id
  ```

  The -businessunit *businessunit id* option is optional when you import the policy file. However, consider the important factors explained in the description of this option later in this topic.

- To import a previously exported file containing one or more policies in disabled status, enter the following command:

  ```
  ixutil import policy -f file -businessunit businessunit id -disable
  ```

Refer to IXUTIL Parameters (see page 14) for more information on the parameters.

3. Log in to CA Service Catalog on the computer where you imported the policies. If you had imported policies into a specific business unit, log in to the respective business unit.

4. Click Catalog, Policies.

   The Policy Builder main page appears.

5. Expand the policy folders. Verify that the Catalog system imported the policy or policies as you want.

6. Close the CA Service Catalog command prompt.

Rollback Invoices Utility

Use the Rollback Invoices command line utility to roll back the most recent invoices for all the accounts defined by invoice group (-g *groupname*). You can optionally limit the rollback to only the accounts in a specific business unit (-b *businessunitid*). This utility runs in batch.

Note: The rollback works *only* for invoices that were created by a successful invoice bill run for the invoice group.

The syntax for using the rollback.bat (Windows) command follows.

rollback -g *groupname* -b *businessunitid*

-g *groupname*

Specifies the name of the invoice group that defines the accounts whose most recent invoices you want to roll back.

Important! This parameter is required.

-b *businessunitid*

Specifies the business unit ID of the accounts--in the named invoice group--whose most recent invoices are being rolled back. If you omit this parameter, the most recent invoices for the accounts in the specified invoice group for *all* business units are rolled back.

For Windows, the rollback.bat file is located in the USM_HOME\accounting\scripts folder.

Examples

The following command rolls back the most recent invoices for all accounts in the InternationalAccounts invoice group, in all business units:

```
rollback -g InternationalAccounts
```

The following command rolls back the most recent invoices for only the accounts in the InternationalAccounts invoice group that belong to the ForwardInc business unit:

```
rollback -g InternationalAccounts -b ForwardInc
```

Data Mediation Aggregation Utility

You can start a data mediation aggregation using a command line utility.

The syntax for using the startDMAggregation.bat (Windows) command is as follows:

```
startDMAggregation.bat [startdate] [enddate]
```

startdate

Specifies the start date in MM/DD/YYYY format for the data to be included in the aggregation.

enddate

Specifies the start date in MM/DD/YYYY format for the data to be included in the aggregation.

For Windows, the startDMAggregation.bat file is located in the USM_HOME\scripts folder.

Chapter 2: Components

CA Service Catalog allows you to deploy and undeploy CA Service Catalog components from the command line.

This section contains the following topics:

Complete the Prerequisites

Before deploying the CA Service Catalog components, you must fulfill the the following prerequisites.

To deploy EEM:

The following columns must be filled in config.properties file.

- eem.configured = true
- eiam.backend =
- eiam.application =
- eiam.admin.username = EiamAdmin
- eiam.admin.password =

To deploy MDB:

The following columns must be filled in config.properties file.

- mdb.configured = true

  Mandatory parameters for command line interface

- mdb.deploy.vendor =
- mdb.deploy.host =
- mdb.deploy.port =
- mdb.deploy.user =
- mdb.deploy.password =
- mdb.deploy.dbinstance =

  Mandatory oracle specific parameter

- mdb.deploy.connectionid =

  MDB Configuration

- mdb.deploy.dbname =
- mdb.deploy.mdbadminpwd =
- mdb.deploy.usmuser =
- mdb.deploy.usmpassword =

  Note: If the database is SQL, then fill only the SQL related columns.

To deploy Component:

- component.catalog = true
- component.accounting = true

- component.webserver = true
- application.service = true
- corporate.service = true
- facilities.service = true
- it.service = true
- network.service = true
- personnel.service = true
- project.service = true
- reservation.service =true

To run all the commands:

All the columns mentioned above must be filled in config.properties.

Note: The Oracle and SQL columns must not be filled at the same time. If the database is Oracle then only the Oracle columns must be filled. The same criteria holds true for SQL database.

Deploy Components

You can deploy the CA Service Catalog components like mdb, eem and component from the command line using the config.properties file. The config.properties file is an input file and must be located outside the USM_HOME. The file accepts passwords in plain text format only. Follow the sequence below to deploy CA Service Catalog components.

Note: You can also deploy CA Service Catalog components from the CA Service Catalog User Interface.

Follow these steps:

1. Open the CA Service Catalog command prompt by clicking Start, Programs, CA, Service Catalog, CA Service Catalog Command Prompt.
2. Go to Scripts path in USM.
3. Enter one or more of the following commands:
   - To deploy mdb enter the following command:

     ```
     Configurator.bat –c mdb –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –c mdb –p "C:\config.properties"
   - To deploy eem enter the following command:

     ```
     Configurator.bat –c eem –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –c eem –p "C:\config.properties"
   - To deploy component enter the following command:

     ```
     Configurator.bat –c component –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –c component –p "C:\config.properties"
   - To deploy all CA Service Catalog components enter the following command:

     ```
     Configurator.bat –c all –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –c all –p "C:\config.properties"

     Important! You must stop and start the tomcat service for the changes to take effect.

You have deployed the CA Service Catalog components using the command line.

Undeploy Components

CA Service Catalog allows you to undeploy the mdb, eem and component individually from the command line. You must undeploy these components in reverse order.

Follow these steps:

1. Open the CA Service Catalog command prompt by clicking Start, Programs, CA, Service Catalog, CA Service Catalog Command Prompt.
2. Go to the Scripts path in USM
3. Enter one or more of the following commands:
   - To undeploy all CA Service Catalog components enter the following command:

     ```
     Configurator.bat –u all –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –u all –p "C:\config.properties"

   - To undeploy component enter the following command:

     ```
     Configurator.bat –u component –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –u component –p "C:\config.properties"

   - To undeploy mdb enter the following command:

     ```
     Configurator.bat –u mdb –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –u mdb –p "C:\config.properties"

   - To undeploy eem enter the following command:

     ```
     Configurator.bat –u eem –p "path to config.properties"
     ```

     For Example: C:\Program Files\ca\service catalog\scripts>Configurator.bat –u eem –p "C:\config.properties"

     Important! You must stop and start the tomcat service for the changes to take effect.

You have undeployed the CA Service Catalog components using the command line.

Appendix A: Java Plug-Ins

This section contains the following topics:

Java Report Plug-Ins

Several built-in Java report plug-ins are provided with CA Service Catalog. The following table displays the sample data objects that are provided with CA Service Catalog that use the built-in Java report plug-ins.

Sample Data Object (Comment)	Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)	Input Parameters (Prompt) / [ConstantValue]	Output Variables
Accounting Component Folder			

Sample Data Object (Comment)	**Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)**	**Input Parameters (Prompt) / [ConstantValue]**	**Output Variables**
Accounting Configuration Service Host (Shows the relationship between services, service option groups, accounts and hosts)	BMS_DbUtilmain	ACCOUNT_LABEL: String (Enter Account Label) OFFERING_NAME :String (Enter Service Name) RATE_PLAN_NAME:String (Enter Service Option Group Name)	account_label, offering_name, rate_plan_name, host_name

Sample Data Object (Comment)	Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)	Input Parameters (Prompt) / [ConstantValue]	Output Variables
Accounting Details	billing.BillingGeneric	DOMAIN:String (Enter Business Unit) INC_CHILD_DOMAINS: Integer (Include Child Business Units (Yes/No)) STATUS:String (Account Status) START_DATE:Date (Account Opened After) END_DATE:Date (Account Opened Before) START_BALANCE:Float (Balance Greater Than Equal To) END_BALANCE:Float (Balance Less Than Equal To) REPORT_TYPE:Constant - [accounting_details]	account_no, account_label, opened_date, status, account_type, aggregate, billing_cycle, billing_cycle_interval, last_invoice_date, billing_cycle_date, invoice_sent_date, balance
Adjustment Details	billing.BillingGeneric	DOMAIN:String (Enter Business Unit) BILL_ADJ_STATUS:Integer (Status) BILL_ADJ_TYPE:Integer (Adjustment Type) BILL_ADJ_VALUE_TYPE:Integer (Adjustment Value Type) BILL_ADJ_GLOBAL:Integer (Global Adjustments) START_DATE:Date (Active After) END_DATE:Date (Active Before) REPORT_TYPE:Constant - [adjustment_details]	adjustment_name, charge_type, adjustment_type, value_type, value, active_date, billing_cycle, period

Sample Data Object (Comment)	Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)	Input Parameters (Prompt) / [ConstantValue]	Output Variables
Invoice Details	billing.BillingGeneric	DOMAIN:String (Enter Business Unit) INC_CHILD_DOMAINS:Integer (Include Child Business Units (Yes/No)) DATE_CHOICE:String (Choose Invoice Date) START_DATE:Date (Date After) END_DATE:Date (Date Before) START_BALANCE:Float (Balance Greater Than Equal To) END_BALANCE:Float (Balance Less Than Equal To) REPORT_TYPE:Constant - [invoice_details]	statement_id, statement_label, invoice_date, due_date, period_from, period_to, balance, violations
Operational Service (Shows a pivot table for services and cost for a business unit)	BillOperational	DOMAIN_PATH:String (Enter Business Unit) START_DATE:Date (Enter Start Date) END_DATE:Date (Enter End Date) USAGE_TYPE:String (Cost/Usage)	offering_name, *

Sample Data Object (Comment)	Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)	Input Parameters (Prompt) / [ConstantValue]	Output Variables
Payment Details	billing.BillingGeneric	DOMAIN:String (Enter Business Unit) INC_CHILD_DOMAINS:Integer (Include Child Business Units) START_DATE:Date (Payment Posted After) END_DATE:Date (Payments Posted Before) START_BALANCE:Float (Payment Greater Than Equal To) END_BALANCE:Float (Payment Less Than Equal To) REPORT_TYPE:Constant - [payment_details]	account_label, status, payment_date, payment_type, amount
Residual Analysis - Monthly	billing.BillPlans	DOMAIN:String (Enter Business Unit) YEAR:Integer (Enter Fiscal Year) SET_ACTIVITY:Integer (Enter Activity) ITEM_KEY:String (Enter Item Key) REPORT_TYPE:Constant - [1] SET_CODE:String (Enter Set Name)	name, start_date, end_date, set_name, actual, budget, pervar
Residual Analysis - Yearly	billing.BillPlans	DOMAIN:String (Enter Business Unit) SET_CODE:String (Enter Set Name) SET_ACTIVITY:Integer (Enter Activity) REPORT_TYPE:Constant - [0] YEAR:Integer (Enter Fiscal Year)	name, set_name, actual, budget, sumvar, enum_4
Catalog Folder			

Sample Data Object (Comment)	**Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)**	**Input Parameters (Prompt) / [ConstantValue]**	**Output Variables**
Request Fulfillment (Shows time taken to Approve, Fulfill, and Complete all requests which fall within the specified date range.)	RequestFulfillmentReport	START_DATE:Date (Enter Start Date) END_DATE:Date (Enter End Date) DOMAIN:String (Enter Business Unit) STATUS_RANGES:Constant - [200-999;999-2;200-2]	request_id, req_desc, status_id, created_date, desired_date, priority, req_by_user_id, req_for_user_id, sys_timestamp, sys_user_id, status_time_millis_1, status_time_millis_2, status_time_millis_3
Request Item Fulfillment (For every request which contains the specified service option row, this shows time taken to Approve, Fulfill, and Complete the request item. Also displays SLA violation data for each request. (sub-report for the Request Item Fulfillment Averages report))	RequestFulfillmentReport	START_DATE:Date (Enter Start Date) END_DATE:Date (Enter End Date) DOMAIN:String (Enter Business Unit) RATE_ITEM_ID:Long (Enter Service Option Element ID) TYPE_ID:Constant - [1] STATUS_RANGES:Constant - [200-999;999-2;200-2]	request_id, offering_name, rate_plan_name, item_text, status_id, sys_timestamp, sys_user_id, status_time_millis_1, wrn_millis_1, vio_millis_1, wrn_variance_millis_1, vio_variance_millis_1, status_time_millis_2, wrn_millis_2, vio_millis_2, wrn_variance_millis_2, vio_variance_millis_2, status_time_millis_3, wrn_millis_3, vio_millis_3, wrn_variance_millis_3, vio_variance_millis_3, wrn_or_vio_1, wrn_or_vio_2, wrn_or_vio_3

Sample Data Object (Comment)	Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)	Input Parameters (Prompt) / [ConstantValue]	Output Variables
Request Item Fulfillment Averages (For each service option row in the system, this report shows a count of how many requests used this row as well as a sum/average of the time taken to Approve, Fulfill, and Complete the items. Also displays SLA violation data for each request item average.)	RequestFulfillmentReport	START_DATE:Date (Enter Start Date) END_DATE:Date (Enter End Date) DOMAIN:String (Enter Business Unit) TYPE_ID:Constant - [2] STATUS_RANGES:Constant - [200-999;999-2;200-2]	item_id, offering_name, rate_plan_name, item_text, total_count, status_time_millis_1, status_time_avg_millis_1, wrn_millis_1, vio_millis_1, wrn_variance_millis_1, vio_variance_millis_1, status_time_millis_2, status_time_avg_millis_2, wrn_millis_2, vio_millis_2, wrn_variance_millis_2, vio_variance_millis_2, status_time_millis_3, status_time_avg_millis_3, wrn_millis_3,vio_millis_3, wrn_variance_millis_3, vio_variance_millis_3, wrn_or_vio_1, wrn_or_vio_2, wrn_or_vio_3

Sample Data Object (Comment)	**Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)**	**Input Parameters (Prompt) / [ConstantValue]**	**Output Variables**
Request Item Fulfillment By Request ID (Shows time taken to Approve, Fulfill, and Complete each request item of the specified request. Also displays SLA violation data for each request item. (sub-report for the Request Fulfillment report))	RequestFulfillmentReport	REQUEST_ID:Long (Enter Request ID) TYPE_ID:Constant - [1] STATUS_RANGES:Constant - [200-999;999-2;200-2]	offering_name, rate_plan_name, item_text,status_id, sys_timestamp, sys_user_id, status_time_millis_1, wrn_millis_1, vio_millis_1, wrn_variance_millis_1, vio_variance_millis_1, status_time_millis_2, wrn_millis_2, vio_millis_2, wrn_variance_millis_2, vio_variance_millis_2, status_time_millis_3, wrn_millis_3, vio_millis_3, wrn_variance_millis_3, vio_variance_millis_3, wrn_or_vio_1, wrn_or_vio_2, wrn_or_vio_3
Financial Reports Folder			
Year-To-Date Actuals by Department	BuService	Business_Report:Constant - [Actuals by Department] Start_Time:Date (Enter Start_Date) End_Time:Date (Enter End_Date) Budget_Set:String (Enter Budget_Set)	Department, Total, Budget_Set, *

Sample Data Object (Comment)	**Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)**	**Input Parameters (Prompt) / [ConstantValue]**	**Output Variables**
Year-To-Date Departmental Variance	BuService	Business_Report:Constant - [Variance by Department] Start_Time:String (Enter Start Date) End_Time:String (Enter End Date) Budget_Set:String (Enter Budget Set)	*, Department, Budget, Actuals, Variance, Budget_Set
Year-To-Date Metric Usage by Department	BuService	Business_Report:Constant - [Metric Usage] Tenant_ID:String (Enter Business Unit) Service_Name:String (Enter Service) Offering_Type:String (Enter Service Type) Start_Time:Date (Enter Start_Date) End_Time:Date (Enter End_Date)	Service Item, Quantity, Rate, Cost, Tenant_ID, Service_Name, *
Year-To-Date Service Usage by Department	BuService	Business_Report:Constant - [Service Usage] Tenant_ID:String (Enter Business Unit ID) Budget_Set:String (Enter Budget Set) Start_Time:Date (Enter Start Date) End_Time:String (Enter End Date)	Service, Budget, Actuals, Variance, Tenant_ID, Budget_Set, Offering_Type, *

Sample Data Object (Comment)	**Java Report Plug-in Class (Prefixed by com.ca.usm.reporting.Plugins)**	**Input Parameters (Prompt) / [ConstantValue]**	**Output Variables**
Year-To-Date Usage of Service by Users	BuService	Business_Report:Constant - [User Usage] Tenant_ID:String (Enter Business Unit ID) Service_Name:String (Enter Service) Unit_Metric:String (Enter Unit Metric) Start_Time:String (Enter Start_Date) End_Time:String (Enter End_Date)	*, Users, Rate, Unit, Cost
Metric Events Folder			
Metric Events (Metric Events System Object)	SMEventFunc	OFFERING_NAME:String (Service Name) START_DATE:Date (Start Date) END_DATE:Date (End Date) offering_id:Constant - [] account_no:Constant - []	account, time, value, Info

Java Rule Action Plug-Ins

Several built-in Java rule action plug-ins are provided with CA Service Catalog. The following table displays the rule actions that are provided with CA Service Catalog that use the built-in Java rule action plug-ins.

Event/Rule/Action	**Java Report Plug-in Class (Prefixed by com.ca.usm. ruleEngine.action)**	**Input Parameters**
User Change/Update eIAM Backend for existing Users /Modifying the eIAM User	CreateEiamUserAction	passwordTemplate (optional) - sets the password of a newly created CA EEM user to the value displayed

User Create/Update eIAM Backend for new Users /Create New eIAM User	CreateEiamUserAction	passwordTemplate (optional) - sets the password of a newly created CA EEM user to the value displayed

Java Scheduler Plug-Ins

Several built-in Java scheduler plug-ins are provided with CA Service Catalog. These are for internal use only.

Appendix B: Event Variables

This section contains the following topics:

Event Variables

Events represent changes standard events that occur in CA Service Catalog components. You can optionally add custom events.

Each standard event occurs when it is caused by an CA Service Catalog action. For example, the User Create event occurs when a new user is added using the Add User user interface or the createUser web service method.

Each event type may have parameters associated with it. When an event occurs, the parameter values reflect the context of the event. For example, when the User Create event occurs, the associated parameter named $user_id$ contains the User ID value for the new user just created.

Event parameter values can be used in rule filters and rule actions.

Account Change Event

This event type occurs when account profile information is changed.

Event Parameter
$account_id$
$closed_date$
$email$
$label$
$location_uuid$
$name$
$opened_date$
$status$
$tenant_id$
$time_zone_code$

Event Parameter
$website$
all

Account Create Event

This event type occurs when an account is created.

Event Parameter
$account_id$
$closed_date$
$email$
$label$
$location_uuid$
$name$
$opened_date$
$status$
$tenant_id$
$time_zone_code$
$website$
all

Adjustment Change Event

This event type occurs when an accounting adjustment is changed.

Note: You must have Accounting Component installed to use this event type.

Event Parameter
$account_no$
$active_date$
$adjustment_id$
$adjustment_name$

Event Parameter
end_date
$inactive_date$
$start_date$
$status$
$value$
all

Adjustment Create Event

This event type occurs when an accounting adjustment is created.

Note: You must have Accounting Component installed to use this event type.

Event Parameter
$account_no$
$active_date$
$adjustment_id$
$adjustment_name$
end_date
$inactive_date$
$start_date$
$status$
$value$
all

Application Added To Account Event

This event type occurs when an account subscribes to or requests a service that includes a service option element of type Application.

Event Parameter
$account_id$
$runtime_application_id$

Application Configuration Change Event

This event type occurs when application configuration is changed.

Event Parameter
$comment$
$runtime_application_id$
$type$
all

Application Configuration Create Event

This event type occurs when an application configuration is created.

Event Parameter
$comment$
$runtime_application_id$
$type$
all

Application Configuration Delete Event

This event type occurs when an application configuration is deleted.

Event Parameter
$runtime_application_id$
all

Application Deleted From Account Event

This event type occurs when an account's subscription to or request for a service that includes a service option element of type Application is unsubscribed or cancelled.

Event Parameter
$account_id$
$runtime_application_id$

Business Unit Change Event

This event type occurs when business unit profile information is changed.

Event Parameter
$closed_date$
$description$
$domain_type$
$opened_date$
$status$
$tenant_id$
$tenant_name$
$time_zone_code$
$website$
all

Business Unit Create Event

This event type occurs when a business unit is created.

Event Parameter
$closed_date$
$description$
$domain_type$
$opened_date$

Event Parameter
$status$
$tenant_id$
$tenant_name$
$time_zone_code$
$website$
all

Catalog Subscription Change Event

This event type occurs when catalog subscriptions change for an account.

Note: You must have Accounting Component installed to use this event type.

Event Parameter
$account_no$
$domain$
$new_offerings$
$new_subscriptions$
$old_offerings$
$old_subscriptions$
$subscription_date$
all

Data Mediation Aggregation Event

This event type occurs when the status for a Data Mediation Aggregation changes.

Event Parameter
end_date
$start_date$
$status$
$status_date$

Data Mediation Error Event

This event type occurs when there is a Data Mediation error.

Event Parameter
$column$
db_id
$destination$
err_code
err_text
$object_id$
$profile_name$
row
$source$
$table_id$
$tran_domain$
$tran_user$
$type$
all

Data Mediation Upload Event

This event type occurs when Data Mediation uploads data.

Event Parameter
db_id
$destination$
$object_id$
$profile_name$
$source$
$status$
$table_id$
$tran_domain$

Event Parameter
$tran_user$
all

Document Change Event

This event type occurs when the definition for a document in document management or the definition for an attachment to a request is modified.

Event Parameter
$asset_id$
$asset_parent_id$
$created_date$
$description$
$file_name$
$file_size$
$icon_used$
is_folder
is_system
$keywords$
$locked$
$lockout_time$
$mime_type$
$modified_date$
$object_id$
$object_type$
$owner$
$parent_tenant_id$
$path$
$tenant_id$
url
$user_id$

Event Parameter
all

Document Create Event

This event type occurs when a document is added to document management or an attachment is added to a request.

Event Parameter
$asset_id$
$asset_parent_id$
$created_date$
$description$
$file_name$
$file_size$
$icon_used$
is_folder
is_system
$keywords$
$locked$
$lockout_time$
$mime_type$
$modified_date$
$object_id$
$object_type$
$owner$
$parent_tenant_id$
$path$
$tenant_id$
url
$user_id$
all

Document Delete Event

This event type occurs when a document is removed from document management or an attachment is removed from a request.

Event Parameter
$asset_id$
$asset_parent_id$
$created_date$
$description$
$file_name$
$file_size$
$icon_used$
is_folder
is_system
$keywords$
$locked$
$lockout_time$
$mime_type$
$modified_date$
$object_id$
$object_type$
$owner$
parent_tenant_id$
$path$
$tenant_id$
url
$user_id$
all

Invoice Change Event

This event type occurs when invoice information is changed.

Note: You must have Accounting Component installed to use this event type.

Event Parameter
$account_no$
$comments$
$credit_amount$
$debit_amount$
due_date
$invoe_date$
$period_from$
$period_to$
$purchase_order_no$
$statement_id$
$statement_label$
$status$
$status_reason$
tax_amount
$violations$
all

Invoice Create Event

This event type occurs when an invoice is generated.

Note: You must have Accounting Component installed to use this event type.

Event Parameter
$account_no$
$comments$
$credit_amount$

Event Parameter
$debit_amount$
due_date
$invoe_date$
$period_from$
$period_to$
$purchase_order_no$
$statement_id$
$statement_label$
$status$
$status_reason$
tax_amount
$violations$
all

Notes Create Event

This event type occurs when a note is added to a request.

Note: You must have CA Service Catalog installed to use this event type.

Event Parameter
$account_id$
$account_label$
$created_date$
$domain$
$note_text$
$note_type$
$notes_id$
ref_1
ref_2
ref_3
$source_id$

Event Parameter
$source_type$
$status$
$type_1$
$type_2$
$type_3$
$user_id$
all

Payment Event

This event type occurs when a payment is made.

Note: You must have Accounting Component installed to use this event type.

Event Parameter
$account_no$
id
$opened_date$
p_name
$type_1$
$type_2$
all

Request Change Event

This event type occurs when request header information is modified.

Note: You must have CA Service Catalog installed to use this event type.

Event Parameter	Meaning	Example
$comments$	Description	My laptop for travel
$completion_date $	Date request is Completed	

Event Parameter	Meaning	Example
$created_date$	Date request is created	
$desired_date$	Date Required	
$domain$	Business Unit Name for the Requested For user or account	ABC Corp Sales
$modified_date$	Date request was last modified	
$name$	Request Name	My new laptop
$priority$	Numeric value for request Priority	3
$req_by_account_id$	Internal ID for Requested By user	10018
$req_by_user_id$	User ID for Requested By user	juser
$req_for_account_id$	Internal ID for account associated with Requested For user or account	10054
$req_for_user_id$	User ID for Requested For user when request is for a user	juser (for a request for a user) NIL (for a request for an account)
$request_id$	Internal ID for request	10023
$status$	Numeric value of status of the request, for example: 400=Pending Approval 800=Approved	400

all	Name=value pairs of all available event parameters including additional data not available as event parameters. Also includes variables containing old data prior to the save causing the event to occur.	comments='My laptop for travel'
		comments_eventdatatype='String'
		comments_old='NIL'
		etc.

Request Create Event

This event type occurs when a request or cart is created.

Note: You must have CA Service Catalog installed to use this event type.

See Request Change Event for details on each parameter.

Event Parameter
$comments$
$completion_date$
$created_date$
$desired_date$
$domain$
$modified_date$
$name$
$priority$
$req_by_account_id$
$req_by_user_id$
$req_for_account_id$
$req_for_user_id$
$request_id$
$status$
all

Request Delete Event

This event type occurs when a request is deleted.

Note: You must have CA Service Catalog installed to use this event type.

See Request Change Event for details on each parameter.

Event Parameter
$comments$
$completion_date$

Event Parameter
$created_date$
$desired_date$
$domain$
$modified_date$
$name$
$priority$
$req_by_account_id$
$req_by_user_id$
$req_for_account_id$
$req_for_user_id$
$request_id$
$status$

Request Item Form Element Change Event

This event type occurs when a field in a form associated with a request item is changed.

Note: You must have Accounting Component installed to use this event type.

Event Parameter	Meaning	Example
$form_elem_name$	Label for field as it appears on the displayed form	Employee Title
$form_elem_value$	Value for field on the form **Note**: This value will be the HTML “value” field which is different from the choice list “label” field displayed.	Vice President (for an input type field) 1 (for a select list type field)
$subscription_detail_id$	Unique internal ID for this request item	10012

Event Parameter	Meaning	Example

all	Name=value pairs of all available event parameters including additional data not available as event parameters. Also includes variables containing old data prior to the save causing the event to occur.	form_elem_label='Employee Title'
		form_elem_label_eventdatatype='String'
		form_elem_label_old='NIL'
		form_elem_name='emp_title'
		form_elem_name_eventdatatype='String'
		form_elem_name_old='NIL'
		etc.

Request Item Form Element Create Event

This event type occurs when a field in a form associated with a request item is initially populated (non-blank).

Note: You must have CA Service Catalog installed to use this event type.

See Request Item Form Element Change Event for details on each parameter.

Event Parameter
$form_elem_name$
$form_elem_value$
$subscription_detail_id$
all

Request/Subscription Item Change Event

This event type occurs when a request or subscription item is modified. The modification is generally a change to the status of the item.

Note: Requests or subscriptions are for service options. Service options are comprised of service option elements. When the status of a service option that is part of a request or subscription changes, this event will occur for each service option element in the service option.

Event Parameter	Meaning	Example
$account_label$	Requested For user's associated Account Name (if requested for a user) or Account Name (if requested for an account or is a subscription)	ABC Corp:juser (for a request for a user) Sales (for a request for an account)
$account_no$	Internal ID for the account in $account_label$	10002
$approval_level$	Numeric value for Approval Level for service	10
$approval_process$	Numeric value for Approval Process for service where: 0=No approval 1=System approval process 2=Workflow driven approval process	2
$category$	Numeric value for requested item category, for example: 0=Software 1=Hardware	0
$category_class$	Numeric value for requested item class within category, for example: For category 0 (Software), class 10 may be “Office”	10
$category_subclass$	Numeric valued for requested item subclass within class within category, for example: For category 0 (Software), class 10 (Office), subclass 10 may be “Microsoft”	10

Event Parameter	Meaning	Example
$charge$	Numeric value indicating whether associated amounts are a charge or credit for use by Accounting Component where: 0=credit 1=charge	1
$charge_date$	Date the item will first be charged for by Accounting Component	
$code$	Service option element Code	My Element Code
$domain$	Business Unit Name for the Requested For user or account	ABC Corp Sales
$enum_1$	Varies depending on service option element type	
$enum_2$	Varies depending on service option element type	
$enum_3$	Varies depending on service option element type	
$enum_4$	Varies depending on service option element type	
$enum_5$	Varies depending on service option element type	
$external_id$	Service option element External ID	My Element External ID
$group_id$	Numeric value indicating the occurrence of the service in a request (in case a request includes more than one copy of a service)	1 NIL (for a subscription)
id	Unique internal ID for this request item	10012
$installments$	Numeric value indicating the number of installments for associated amounts for use by Accounting Component	0
$instance_name$	Optional value assigned when subscribing an account to a service	My instance
$item_id$	Internal ID for requested service option element	9981

Event Parameter	Meaning	Example
$last_charge_date$	Date the item was last charged for by Accounting Component	
$numeric_1$	Varies depending on service option element type	
$numeric_2$	Varies depending on service option element type	
$offering_id$	Internal ID for requested service	10002
$offering_name$	Service Name	Standard Laptop Bundle
$rate_item_col$	Numeric value (zero-based) indicating the associated service option element's column position in the service option row	0
$rate_item_name$	Service option element Display Text	Standard Laptop
$rate_item_row$	Numeric value (one-based) indicating the associated service option element's row position in the service option group	1
$rate_item_text_1$	Additional text associated with the service option element. For example, for a Text type service option element, this is Text Value and for a Rate type service option element, this is Display Unit Type.	/month

Event Parameter	Meaning	Example
$rate_item_type$	Numeric value for the associated service option element where: 0=Text 1=Header 2=Numeric Range 3=Rate 4=Application 5=Agreement 6=Numeric 7=Boolean 8=Adjustment 9=Date 10=Date Range 11=Day 12=Allocation 13=Form	3
$rate_plan_id$	Internal ID for associated service option group	10035
$rate_plan_name $	Associated service option group Name	Standard Laptop Accessories
$req_by_user_id$	User ID for Requested By user	Juser NIL (for a subscription)
$req_for_user_id$	User ID for Requested For user when request is for a user	juser (for a request for a user) NIL (for a request for an account) NIL (for a subscription)
$request_id$	Internal ID for request	10023 (for a request) NIL (for a subscription)
$request_name$	Request Name	My new laptop NIL (for a subscription)
$request_priority $	Numeric value for request Priority	3 NIL (for a subscription)
sd_row	Numeric value (one-based) for the service option containing this service option element relative to all the service options selected with the service	1

Event Parameter	Meaning	Example
$status$	Numeric value of status of the requested item, for example: 400=Pending Approval 800=Approved	400
$subscribed_date$	Date the item was included in a request or subscription	
$subscription_type$	Numeric value indicating the type of item where: 0=Other 1=Application 2=Agreement 3=Adjustment 4=Form	0
$text_1$	Varies depending on service option element type	
$text_2$	Varies depending on service option element type	
$text_3$	Varies depending on service option element type	
$text_4$	Varies depending on service option element type	
$text_5$	Varies depending on service option element type	
$text_6$	Varies depending on service option element type	
$text_7$	Varies depending on service option element type	
$tiered_item_id$	Internal ID of the tiered service option group of the associated service option element where: -1=Not associated with a tiered service option group	-1

Event Parameter	Meaning	Example
$tiered_last_date$	Date the item was last charged for by Accounting Component. Applicable only if the related service option element is part of a tiered service option group.	
$track_as_asset$	Numeric value indicating whether this request item should be tracked as an asset where: 0=No 1=Yes	1
$unsubscribed_date$	Date the item is cancelled or unsubscribed	NIL

all	Name=value pairs of all available event parameters including additional data not available as event parameters. Also includes variables containing old data prior to the save causing the event to occur.	account_label='Sales'
		account_label_eventdatatype='String'
		account_label_old='Sales'
		account_no='10003'
		account_no_eventdatatype='String'
		account_no_old='10003'
		etc.

Request/Subscription Item Create Event

This event type occurs when a request or subscription item is created.

Note: Requests and subscriptions are for service options. Service options are comprised of service option elements. When a request or subscription for a service option is made, this event will occur for each service option element in the service option.

See Request/Subscription Item Change Event for details on each parameter.

Event Parameter
$account_label$
$account_no$
$approval_level$
$approval_process$
$category$
$category_class$

Event Parameter
$category_subclass$
$charge$
$charge_date$
$code$
$domain$
$enum_1$
$enum_2$
$enum_3$
$enum_4$
$enum_5$
$external_id$
$group_id$
id
$installments$
$instance_name$
$item_id$
$last_charge_date$
$numeric_1$
$numeric_2$
$offering_id$
$offering_name$
$rate_item_col$
$rate_item_name$
$rate_item_row$
$rate_item_text_1$
$rate_item_type$
$rate_plan_id$
$rate_plan_name$
$req_by_user_id$
$req_for_user_id$
$request_id$

Event Parameter
$request_name$
$request_priority$
sd_row
$status$
$subscribed_date$
$subscription_type$
$text_1$
$text_2$
$text_3$
$text_4$
$text_5$
$text_6$
$text_7$
$tiered_item_id$
$tiered_last_date$
$track_as_asset$
$unsubscribed_date$
all

Request/Subscription Item Delete Event

This event type occurs when a request or subscription item is deleted.

Note: Requests and subscriptions are for service options. Service options are comprised of service option elements. When a request or subscription for a service option is deleted, this event will occur for each service option element in the service option.

See Request/Subscription Item Change Event for details on each parameter.

Event Parameter
$account_label$
$account_no$
$approval_level$

Event Parameter
$approval_process$
$category$
$category_class$
$category_subclass$
$charge$
$charge_date$
$code$
$domain$
$enum_1$
$enum_2$
$enum_3$
$enum_4$
$enum_5$
$external_id$
$group_id$
id
$installments$
$instance_name$
$item_id$
$last_charge_date$
$numeric_1$
$numeric_2$
$offering_id$
$offering_name$
$rate_item_col$
$rate_item_name$
$rate_item_row$
$rate_item_text_1$
$rate_item_type$
$rate_plan_id$
$rate_plan_name$

Event Parameter
$req_by_user_id$
$req_for_user_id$
$request_id$
$request_name$
$request_priority$
sd_row
$status$
$subscribed_date$
$subscription_type$
$text_1$
$text_2$
$text_3$
$text_4$
$text_5$
$text_6$
$text_7$
$tiered_item_id$
$tiered_last_date$
$track_as_asset$
$unsubscribed_date$
all

Service Change Event

This event occurs when the service or folder details, such as the name, are changed.

Event Parameter
$date_available$
$date_cancelled$
$date_created$
$date_unavailable$

$description$
$offering_id$
$offering_name$
$status$
all

Service Create Event

This event occurs when a service or folder is created.

Event Parameter
$date_available$
$date_cancelled$
$date_created$
$date_unavailable$
$description$
$offering_id$
$offering_name$
$status$
all

Service Option Group Change Event

This event occurs when a service option group is changed either by changing its detail information, such as its name, or by changing its service option elements.

Event Parameter
$date_available$
$date_cancelled$
$date_created$
$date_unavailable$
$description$
$is_inherited$

Event Parameter
$rate_plan_id$
$rate_plan_name$
$rate_plan_type$
$status$
$total_cols$
$total_rows$
all

Service Option Group Create Event

This event occurs when a service option group is created.

Event Parameter
$date_available$
$date_cancelled$
$date_created$
$date_unavailable$
$description$
$is_inherited$
$rate_plan_id$
$rate_plan_name$
$rate_plan_type$
$status$
$total_cols$
$total_rows$
all

System Alerts Event

This event occurs when a system alert is generated.

Event Parameter
$host_name$
$message$
$message_type$
$other_info1$
$other_info2$
$other_info3$
$severity$
$source_name$
$time_stamp$
$username$
all

User Added to Account Event

This event occurs when a user is added to an account.

Event Parameter
$account_id$
$user_id$

User Change Event

This event occurs when user profile information is changed.

Event Parameter
$country$
$default_domain$
$time_zone_code$
$title$

Event Parameter
uid
$user_id$
all

User Create Event

This event occurs when a new user is created.

Event Parameter
$country$
$default_domain$
$time_zone_code$
$title$
uid
$user_id$
all

User Delete Event

This event occurs when a user is deleted.

Event Parameter
$user_id$
all

User Deleted From Account Event

This event occurs when a user is removed from an account.

Event Parameter
$account_id$
$user_id$

User Lockout Event

This event occurs when a user is locked out.

Event Parameter
$user_id$