

CA Service Catalog

Reference Guide

Release 12.7.00



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.

Copyright © 2011 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

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

- CA Service Catalog, which includes CA Service Accounting
- CA Embedded Entitlements Manager (CA EEM)
- CA Automation Suites (formerly CA Spectrum Automation Manager)
- CA Automation Suites Reservation Manager (Reservation Manager)
- CA Business Service Insight (CA BSI, formerly CA Oblicore Guarantee)
- 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 you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following:

- 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.

If you would like 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	7
IXUTIL Import/Export Utility	7
IXUTIL Files	8
Data File Types	8
IXUTIL Export Syntax	9
IXUTIL Import Syntax	10
Object Syntax	10
Object Syntax Rules and Considerations	13
IXUTIL Parameters	14
Object Modification	17
Object Modification Considerations	18
Date Last Modified Conditions	18
Migration for All Objects or Specified Objects	19
Export a Service Using IXUTIL Example	22
Export a Service Option Group Using IXUTIL Example	23
Export and Import Events, Rules, and Actions	25
Import and Export Payments Using IXUTIL Example	28
Import Catalog Entries from a CSV File Using IXUTIL	32
Import CA Workflow Forms into the Form Designer	35
Import and Export Form Designer Forms	37
Import and Export Form Designer Forms Associated to Service Option Elements	39
Export and Import Policies	41
IXCATALOG Import/Export Utility	43
IXCATALOG Syntax	43
Rollback Invoices Utility	44
Data Mediation Aggregation Utility	45
Appendix A: Java Plug-Ins	47
Java Report Plug-Ins	47
Java Rule Action Plug-Ins	55
Java Scheduler Plug-Ins	56
Appendix B: Event Variables	57
Event Variables	57
Account Change Event	57
Account Create Event	58

Adjustment Change Event.....	58
Adjustment Create Event	59
Application Added To Account Event.....	59
Application Configuration Change Event	60
Application Configuration Create Event.....	60
Application Configuration Delete Event.....	60
Application Deleted From Account Event	61
Business Unit Change Event	61
Business Unit Create Event	61
Catalog Subscription Change Event	62
Data Mediation Aggregation Event.....	62
Data Mediation Error Event	63
Data Mediation Upload Event.....	63
Document Change Event.....	64
Document Create Event	65
Document Delete Event	66
Invoice Change Event	67
Invoice Create Event	67
Notes Create Event	68
Payment Event	69
Request Change Event	69
Request Create Event.....	71
Request Delete Event.....	71
Request Item Form Element Change Event	72
Request Item Form Element Create Event.....	73
Request/Subscription Item Change Event	74
Request/Subscription Item Create Event.....	79
Request/Subscription Item Delete Event.....	81
Service Change Event.....	83
Service Create Event	84
Service Option Group Change Event.....	84
Service Option Group Create Event	85
System Alerts Event	86
User Added to Account Event	86
User Change Event	86
User Create Event	87
User Delete Event	87
User Deleted From Account Event.....	87
User Lockout Event	88

Chapter 1: Utilities

CA Service Catalog provides several command line utilities.

This section contains the following topics:

[IXUTIL Import/Export Utility](#) (see page 7)

[IXCATALOG Import/Export Utility](#) (see page 43)

[Rollback Invoices Utility](#) (see page 44)

[Data Mediation Aggregation Utility](#) (see page 45)

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) and rate plans (service option groups) from a CA Service Catalog r12 system and import them as services and service option groups into a CA Service Catalog Release 12.6 system, using the same XML structure.

You can use IXUTIL to migrate new and updated objects from one computer to another. 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 *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 the same XML format must be used in the data files when importing. For some object types, data can also be imported in a proprietary CSV (comma-separated value) format. 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
 - Service Option Groups
 - Services
 - Without the definition (related service option groups)
 - With the definition
 - With Service Level Agreement information
- Business Units
 - Accounts
- Reporting components
 - Data objects
 - Data views
 - Layouts
 - Variables

- Administrative components
 - Events, rules and actions
 - Data mediation profiles
 - request SLA, business (service) hours, and outage calendars
 - Forms
 - Policies
- Data from a database table
- CA Service Catalog data
 - Requests
- CA Service Accounting data
 - Statements
 - Adjustments
 - Transactions
 - Payments
 - Cost pools
 - Plan data
 - Plan set
 - Subscription Details

IXUTIL Export Syntax

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

```
ixutil export object [-f filename] [-format xml] [-domain businessunitname] [object-specific parameters]
```

object

Specifies one of the data types.

-f *filename*

Specifies the name of the file to be exported to; if omitted, the data is exported to the command window; if no folder path is provided, only a file name, the folder used is %USM_HOME%\view\exports.

-format xml

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

-domain *businessunitname*

Specifies the name of the business unit for the information to be exported; this parameter is not always relevant for the object specified.

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 filename [-format xml|csv] [-domain businessunitname] [object-specific parameters]
```

object

Specifies one of the objects, using the [object syntax](#) (see page 10).

-f *filename*

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.

-format *xml|csv*

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

-domain *businessunitname*

Specifies the name of the business unit for the information to be imported. This parameter applies to certain objects 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 accounts in one command, events in another command, rate plans (service option groups) in a third command, and so forth. You cannot combine two or more types of objects in a single import command. An exception is the offeringrateplan option, 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.

ixutil import account [-f file] [-format xml] [-domain id]

ixutil export account [-f file] [-domain id]

ixutil import adjustment [-f file] [-format xml]

ixutil export adjustment [-f file] [-domain id]

ixutil import costpool [-f file] [-format xml]

ixutil export costpool [-f file] [-domain id]

ixutil import datamediationprofile [-f file] [-format xml]

ixutil export datamediationprofile [-f file] [-profile name]

ixutil import event [-f file] [-event name] [-rule name] [-action name]

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

ixutil import form [-f file] [-format xml] [-cawf -domain id]

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

ixutil import generic [-f file] [-format xml] [-table name]

ixutil export generic [-f file] [-format xml|csv|sql] [-table name]

ixutil import offering [-f file] [-format xml] [-domain id] [-include_incomplete_sub] [-include_cmdb_ci_mapping] [-include_permissions] [-include_request_sla] [-include_service_hour]

ixutil export offering [-f file] [-domain id] [-date_last_modified date] [-include_cmdb_ci_mapping] [-include_permissions] [-include_request_sla] [-include_service_hour]

ixutil import offeringrateplan [-f file] [-format xml] [-folder] [-include_incomplete_sub] [-include_cmdb_ci_mapping] [-include_permissions] [-include_request_sla] [-include_service_hour]

ixutil export offeringrateplan [-f file] [-folder name][[-offering name] [-include_forms] [-date_last_modified date] [-include_cmdb_ci_mapping] [-include_permissions] [-include_request_sla] [-include_service_hour]

ixutil import offeringrateplan_csv [-f file] [-domain id]

Note: The offeringrateplan option includes offerings, rate plans, and rate items (service option elements).

ixutil import payments [-f file] [-format csv|xml]

ixutil export payments [-f file] [-for accounts] [-from date] [-to date]

ixutil import plandata [-f file] [-format xml]

ixutil export plandata [-f file] [-domain id]

ixutil import planset [-f file] [-format xml]

ixutil export planset [-f file] [-domain id]

ixutil import rateplan [-f file] [-format xml] [-domain id] [-include_incomplete_sub]

ixutil export rateplan [-f file] [-domain id] [-date_last_modified date] [-include_forms]

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

ixutil export report_data [-f file] [-folder name]

ixutil import report_dataview [-f file] [-format xml] [-folder name]

ixutil export report_dataview [-f file] [-folder name]

ixutil import report_layout [-f file] [-format xml] [-folder name]

ixutil export report_layout [-f file] [-folder name]

ixutil import report_variable [-f file] [-format xml]

ixutil export report_variable [-f file] [-domain id]

ixutil import request [-f file] [-format xml]

ixutil export request [-f file] [-domain id]

ixutil import requestslaservicehour [-f file] [-format xml] [-domain id]

ixutil export requestslaservicehour [-f file] [-domain id]

ixutil import slaoffering [-f file] [-format xml]

ixutil export slaoffering [-f file] [-domain id]

ixutil import statement [-f file] [-format xml] [-domain id]

ixutil export statement [-f file] [-domain id]

ixutil import subscription_detail [-f file] [-format xml]

ixutil export subscription_detail [-f file] [-domain id]

ixutil import tenant [-f file] [-format xml] [-domain id]

ixutil export tenant [-f file] [-domain id]

ixutil import transaction [-f file] [-format xml]

ixutil export transaction [-f file] [-domain id]

Notes:

- Parameter values that contain spaces must be in double quotes.
- For lists, values are separated by commas.
- Date values must use the format YYYY/MM/DD H24:MM:SS.
- 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 tenant -f output.xml -domain “CA”`.
- The “offering” and “rateplan” references are to old terminology from prior releases. The current terminology for “offering” is “service” and for “rateplan” is “service option group”. For example, to export all the services for the “CA” business unit, use: `ixutil export offering -f output.xml -domain “CA”`.
- The slaoffering object defines request SLA packages. A catalog service option element can be tied to a request SLA package. The slaoffering object must be imported prior to importing any catalog elements that use request SLA packages.
- Report-related objects must be imported in a particular order to support interdependencies:
 - report_variable
 - report_data
 - report_dataview
 - report_layout

Object Syntax Rules and Considerations

The following rules apply to the ixutil command line:

- 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.
- Report-related objects must be imported in the following order to support interdependencies:
 1. report_variable
 2. report_data
 3. report_dataview
 4. report_layout

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 tenant -f output.xml --domain "CA"
```

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

```
ixutil export offering -f output.xml --domain "CA"
```

- The slaoffering object defines request SLA packages. A catalog service option element can be tied to a request SLA package. The slaoffering object must be imported prior to importing any catalog elements that use request SLA packages.

IXUTIL Parameters

The following parameters are used for the IXUTIL command:

Domain

Specifies the Business Unit. For exporting, when the domain parameter is used, the business unit specified must exist in the source system. For importing a business unit, the business unit’s parent business unit should exist in the target system. If it does not, the business unit being imported will become a child business unit of the root business unit.

When you import a business unit that has one or more sub-business units (“tenants”), the order of import *must* be the following:

1. All tenants
2. All accounts
3. All account-related components, such as subscriptions, invoices, adjustments, and so forth

Important! If you do not follow this order of import, the sub-business units, their accounts, and related components will *not* be imported properly and will *not* be matched correctly with the parent business unit!

Event

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 information about the specified event type.

Table

Specifies a database table name. This parameter is used with the “generic” option and will import or export a specific table. This option should be used carefully as many tables have dependencies.

Folder for service-related objects

Specifies a folder or service name in the catalog hierarchy. This option can be used to export a sub-tree of the services hierarchy or identify the parent folder for imported data. When importing, if the parent folder does not exist, the root folder will be used.

Folder for report-related objects

For the report-related objects, specifies the folder parameter that can be used to export a subset of the tree or identify the parent folder for imported data.

Format XML

Specifies that XML format should be used for import or export files. By default, XML is the format 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 checks for subscriptions in all services, including inherited services, child services, and inherited services of child services.

-date_last_modified "*date time*"

Exports only the objects (such as services and service option groups) that are added, defined, edited, or deleted *on or after* this date and time. Specify this option when exporting offerings (services) or rate plans (service option groups).

The terms *added*, *defined*, *edited*, and *deleted* mean any of the following:

- A folder, service, service option element, or subscription has been created (added), deleted, or defined
- A folder or service has had its properties edited
- A service option element or subscription has been added to or deleted from a service
- A service option element or subscription has been moved from one service to another

- A service has been added to or deleted from a folder
- A service has been moved from one folder to another
- The properties of a service or folder have been edited

If you do not specify `-date_last_modified`, all objects are exported.

To specify the date and time, use the format `yyyy/mm/dd mm:ss`. Specify the time in Greenwich Mean Time (GMT).

For example, to export only objects that are created on or after April 29, 2009 (any time during the day), specify the following in the `ixutil export` command:

```
... -date_last_modified "2009/04/29 00:00" ...
```

-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 offering or offeringrateplan commands.

Sample commands follow:

```
ixutil export offeringrateplan -f test_sla.xml -offering blackberry -include_request_sla -include_service_hour
```

```
ixutil import offeringrateplan -f C:\ca\slcm\trunk\build\install\view\exports\test_sla.xml -format xml  
-include_request_sla -include_service_hour
```

-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 offering or offeringrateplan commands.

Sample commands follow:

```
ixutil export rateplan -f test_sla_rp.xml -domain ca.com
```

```
ixutil import rateplan -f C:\ca\slcm\trunk\build\install\view\exports\test_sla_rp.xml -domain ca.com
```

```
ixutil export offering -f test_sla.xml -domain ca.com -include_request_sla -include_service_hour
```

```
ixutil import offering -f C:\ca\slcm\trunk\build\install\view\exports\test_sla.xml -domain ca.com  
-include_request_sla -include_service_hour
```


requestslaservicehour

Exports or imports request SLA service hours (outage calendars with associated outage groups and outages, and business hours) as stand-alone objects, without associated services (offerings).

Sample commands follow:

```
ixutil export requestslaservicehour -f servicehour.xml -domain ca.com
```

```
ixutil import requestslaservicehour -f C:\ca\slcm\trunk\build\install\view\exports\servicehour.xml -domain ca.com
```

form-related parameters

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

- `-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 35)
- [Import and export Form Designer forms](#) (see page 37)
- [Import and export Form Designer forms associated to service option elements](#) (see page 39)

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.

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

Migration for All Objects or Specified Objects

You can specify whether IXUTIL exports and imports *all* objects or *only* objects that meet certain criteria, such as using the `-folder` option to export only the services of the specified folder, as follows:

- The `-offering` option exports only the services grouped under the specified service.
- The `-date_last_modified` option exports only the services updated since the date and time you specify.

Migration is determined by the following:

- The date-time stamps of any same-named objects on the source and destination computers
- Whether you export all objects or only those whose time-date stamps are newer than existing objects (`-date_last_modified` option)
- Whether you import objects that belong to incomplete subscriptions (`-include_incomplete_sub` option)

You can import or export offerings and rate plans in a single command or in two separate commands. The only difference between these options is the size of the XML file generated: the offering XML file alone or the rate plan XML file alone is proportionally smaller than the size of the combined XML file that contains both offerings and rate plans.

Export and Import All Objects

You can export all objects to the target computer from an existing CA Service Catalog implementation. New and updated objects are received on the target computer. If a same-named object on the target computer is older than the object on the source computer, the version from the source computer overwrites the earlier version on the target computer. You can export all objects in the following ways:

- From a command prompt on the source computer, specify the following commands, in the order shown:

```
ixutil export rateplan -f rateplans.xml
ixutil export offering -f offerings.xml
```

From a command prompt on the target computer, specify the following commands, in the order shown:

```
ixutil import rateplan -f "pathname\rateplans.xml"
ixutil import offering -f "pathname\offerings.xml"
```

pathname

Specifies the path name of the folder where the XML file which needs to be imported is located. If the path name contains any spaces, enclose it in quotation marks. Examples:

- "C:\Program Files\CA\Service Delivery\view\rateplans.xml" for rate plans
- "C:\Program Files\CA\Service Delivery\view\offerings.xml" for offerings
- Export and import offerings and *associated* rate plans simultaneously in a single command by using the offeringrateplan option, by entering the following commands on the source and target computers, respectively:

```
ixutil export offeringrateplan -f offeringrateplans.xml
ixutil import offeringrateplan -f "pathname\offeringrateplans.xml"
```

Only rate plans that are associated to an offering are exported and imported. Rate plans that are not associated to an offering are neither exported nor imported.

All new objects from the source computer are written to the target computer. All existing same-named objects on the target computer are overwritten by any newer versions from the source computer. Other same-named objects on both computers are not affected by the IXUTIL command.

Export and Import Changes Made Since a Specific Date and Time

You can export only new and updated objects to the production computer from a large existing CA Service Catalog implementation. Using the `-date_last_modified` option, you want to export only objects that have been created or updated since your last export-import operation, which was performed on May 30, 2008 at 09:08:44 GMT. You can export only objects that have been created or updated since your last export-import operation in the following ways:

- From a command prompt on the source computer, specify the following commands:

```
IXUtil export rateplan -f rateplans.xml -date_last_modified "date time"
IXUtil export offering -f offerings.xml -date_last_modified "date time"
```

date

Specifies the date in yyyy/mm/dd format, in this example, 2008/05/30.

time

Specifies the time date in hh:mm:ss format, in this example, 09:08:44.

If you specify the time, you must enclose the entire date and time expression in quotation marks. In this example, you specify the following:

```
IXUtil export rateplan -f rateplans.xml -date_last_modified "2008/05/30 09:08:44"
```

From a command prompt on the target computer, specify the following command:

```
IXUtil import rateplan -f "pathname\rateplans.xml"
IXUtil import offering -f "pathname\offerings.xml"
```

pathname

Specifies the path name of the folder to which you are importing. If the path name contains any spaces, enclose it in quotation marks. Examples:

- "C:\Program Files\CA\Service Delivery\view\rateplans.xml" for rate plans
- "C:\Program Files\CA\Service Delivery\view\offerings.xml" for offerings
- You can export and import offerings and *associated* rate plans simultaneously in a single command by using the `offeringrateplan` option, by entering the following commands on the source and target computers, respectively:

```
IXUtil export offeringrateplan -f offeringrateplans.xml -date_last_modified "2008/05/30 09:08:44"
IXUtil import offeringrateplan -f "pathname\offeringrateplans.xml"
```

Only rate plans that are associated to an offering are exported and imported. Rate plans that are not associated to an offering are neither exported nor imported.

All objects that have been updated or created since May 30, 2008 at 09:08:44 GMT are exported from the source computer and imported to the target computer.

Preservation of Inheritance

When you export services, 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 command *in sequence* to export the services with inheritance preserved:

```
ixutil export offeringrateplan -f output.xml -offering A,B --domain BU-x
```

```
ixutil import offeringrateplan -f output.xml --domain 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 offeringrateplan -f output1.xml -offering A --domain BU-x
```

```
ixutil export offeringrateplan -f output2.xml -offering B --domain BU-x
```

```
ixutil import offeringrateplan -f output1.xml --domain BU-y
```

```
ixutil import offeringrateplan -f output2.xml --domain BU-y
```

Export a Service Using IXUTIL Example

This example shows how IXUTIL can import or export the offering (service) object in XML format only.

To export a service named “Service 1”

1. Open a command window.
2. Navigate to the %USM_HOME%\scripts folder.
3. Enter the following command:

```
ixutil export offering -f output.xml --folder “Service 1” --domain “CA”
```

The definition of the service named “Service 1” in the business unit named “CA” is exported to the %USM_HOME%\view\exports\output.xml file.

The format of the exported output.xml file is shown below.

```
<?xml version="1.0" ?>
<xml>
  <offering>
    <name>Service 1</name>
    <domain>CA</domain>
    <status>AVAILABLE</status>
    <date_created>2002/04/19 04:01:00</date_created>
    <date_available>2002/04/19 04:01:00</date_available>
    <code>X</code>
    <selection_type>MULTIPLE</selection_type>
    <is_leaf>1</is_leaf>
    <sort_order_type>0</sort_order_type>
    <sort_order_number>2</sort_order_number>
    <sort_order>0</sort_order>
    <rate_plan>
      <domain>CA</domain>
      <name>Service Option Group 1</name>
      <date_created>2002/04/19 04:01:00</date_created>
      <selected_row>1</selected_row>
    </rate_plan>
  </offering>
</xml>
```

Export a Service Option Group Using IXUTIL Example

This example shows how IXUTIL can import or export the rateplan (service option group) object in XML format only.

To export a service option group named "Service Option Group 1"

1. Open a command window.
2. Navigate to the %USM_HOME%\scripts folder.
3. Enter the following command:

```
ixutil export rateplan -f output.xml -folder "Service Option Group 1"
-domain "ca.com"
```

The definition of the service option group named "Service Option Group 1" in the business unit named "ca.com" is exported to the %USM_HOME%\view\exports\output.xml file.

The format of the exported output.xml file is shown below.

```
<?xml version="1.0" ?>
<xml>
  <rate_plan>
    <name>Service Option Group 1</name>
    <domain>ca.com</domain>
    <status>AVAILABLE</status>
    <date_created>2002/04/19 04:01:00</date_created>
    <date_available>2002/04/19 04:01:00</date_available>
    <type>FIXED</type>
    <selection_type>SINGLE</selection_type>
    <has_header>0</has_header>
    <is_inherited>0</is_inherited>
    <total_rows>2</total_rows>
    <total_cols>1</total_cols>
    <rate_item>
      <item_type>3</item_type>
      <item_text>charge</item_text>
      <row>1</row>
      <col>0</col>
      <status>ACTIVE</status>
      <is_hidden>0</is_hidden>
      <currency_1>0</currency_1>
      <date_available>2002/04/19 18:20:50</date_available>
      <unit_cost>0</unit_cost>
      <unit_type>unit</unit_type>
      <billing_cycle>ONE_TIME</billing_cycle>
      <periodic_type>DAILY</periodic_type>
      <periodic_type_interval>-1</periodic_type_interval>
      <installments>1</installments>
      <charge_type>DEBIT</charge_type>
      <cost_type>USER</cost_type>
      <quantity_type>FLAT_RATE</quantity_type>
      <quantity_value>1.0</quantity_value>
      <show_quantity>1</show_quantity>
    </rate_item>
  </rate_plan>
</xml>
```


Export and Import Events, Rules, and Actions

You can use `ixutil` to migrate new and updated objects, including events, rules, and actions, from one computer to another. If necessary, you export and import events, rules, and actions when you upgrade or migrate CA Service Catalog. You also export and import events, rules, and actions when you upgrade your computer or if a computer fails.

Follow these steps:

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


```
ixutil export event -f filename -format xml
```
 - To export all events in a business unit, enter the following command. The command exports all rules and actions in each event.


```
ixutil export event -f filename -format xml -domain name
```
 - To export an event, use the following command. Optionally add the `-domain` parameter from the previous bullet to specify a business unit. The command exports the specified event, including all its rules and actions.


```
ixutil export event -event eventname -f filename -format xml
```
 - To export an event with only the specified rule, use the following command. Optionally add the `-domain` parameter to specify a business unit. The command exports the specified event, with the specified rule and all its actions.


```
ixutil export event -event eventname -rule rulename -f filename -format xml
```
 - To export an event with only the specified rule and the specified action, use the following command. Optionally add the `-domain` parameter to specify a business unit. The command exports the specified event, with the specified rule and the specified action.


```
ixutil export event -event eventname -rule rulename -action actionname -f filename -format xml
```

The following options may require explanation:

-f filename

Specifies the name of the file being exported or imported. This file contains the objects (events, rules, or actions) that you are exporting or importing.

The Catalog system uses an XML file to export and import these objects.

-event *eventname* -rule *rulename* -action *actionname*

Specifies the name of the event, rule, or 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"
```

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.

-domain *name*

Specifies the business unit that contains the objects being exported or imported.

If you are exporting all events in all business units, omit this parameter; otherwise, this parameter is required.

If you are importing objects, optionally specify this parameter to import all the objects into a single business unit, as follows:

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

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

3. Enter one or more of the following import commands on the target computer, as applicable:

- To import all events in all business units, enter the following command. The command imports all rules and actions in each event.

```
ixutil import event -f filename
```

- To import all events in a business unit, enter the following command. The command imports all rules and actions in each event.

```
ixutil import event -f filename -domain business unit name
```

- 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 filename
```

- To import an event with only the specified rule, use the following command. Optionally add the `-domain` parameter to specify a business unit. The command imports the specified event, with the specified rule and all its actions.

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

- To import an event with only the specified rule and the specified action, use the following command. Optionally add the `-domain` parameter to specify a business unit. The command imports the specified event, with the specified rule and the specified action.

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

- 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 filename -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 filename -disable
```

The following options may require explanation:

-disable

Imports the events, rules, and actions in disabled status.

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

When you use the `-disable` option, all objects (events, rules, and actions) that you specify in the command are imported as disabled.

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

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

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

5. Log in to CA Service Catalog on the computer where you imported the objects. If applicable, log in to the target business unit.
6. Click Administration, Tools, Events-Rules-Actions.
The Event Types page appears.
7. View the events, including the rules and actions. Verify that the Catalog system imported the objects as you wanted.
8. Close the CA Service Catalog command prompt.

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

Import and Export Payments Using IXUTIL Example

In this example, IXUTIL imports and exports payments. Payments can be imported into CA Service Accounting by statement, by account, or by department.

To use the tool, run IXUTIL from the command line. Full syntax is displayed when you run the command without any parameters.

```
ixutil export payments [-f file] [-for accounts] [-from date] [-to date]
```

Example: Export All Payments Made After July 1st 2004

This example shows the command line syntax to export all payments made after July 1st 2004:

```
ixutil export payments out.xml -from "2004/07/01"
```

To import payments:

```
ixutil import payments in.xml -format xml
```

Payments are exported into and imported from flat files. The currently supported file formats are XML and CSV (for import). Do not include field headers in the CSV file. For CSV files, fields must be listed in the order previously listed. For XML files, all the tags must be present, but can be blank. The following fields/tags are used for the import/export:

Credit Card Payments

- account_label
- statement_id
- statement_label
- amount
- payment_date
- payment_method
- comments
- pm_status
- pm_type
- pm_issued_by
- pm_card_number
- pm_expiration_date
- pm_holder_name
- pm_holder_address1
- pm_holder_address2
- pm_holder_city
- pm_holder_state
- pm_holder_zip
- pm_holder_country
- pm_holder_telephone
- pm_opened_date

- pm_closed_date
- pm_active_date
- pm_inactive_date
- pm_comments

Prepay Payment Method

- account_label
- statement_id
- statement_label
- amount
- payment_date
- payment_method
- comments
- pm_status
- pm_deposited_amount
- pm_applied_amount
- pm_opened_date
- pm_closed_date
- pm_active_date
- pm_inactive_date
- pm_comments

Other Payment Methods

- account_label
- statement_id
- statement_label
- amount
- payment_date
- payment_method
- comments
- pm_status
- pm_bank_routing_no
- pm_checking_account_no
- pm_check_no

- pm_inst_name
- pm_inst_address1
- pm_inst_address2
- pm_inst_city
- pm_inst_state
- pm_inst_zip
- pm_inst_country
- pm_inst_telephone
- pm_inst_fax
- pm_inst_email
- pm_opened_date
- pm_closed_date
- pm_active_date
- pm_inactive_date
- pm_comments

Different payment methods can be mixed in the same flat file. The fields take the following values:

- payment_method:
 - "Credit Card"
 - "Check"
 - "Cash"
 - "Prepay"
- pm_status:
 - -1 (inactive)
 - 0 (inactive)
 - 1 (active)
 - 2 (opened)
 - 3 (closed)
 - 4 (deleted)

Note: Prepayment credits that have not been applied to statements will not be exported.

Import Catalog Entries from a CSV File Using IXUTIL

The IXUTIL offeringrateplan_csv object can import services, service option groups, and service option elements from a comma-separated value (CSV) file. The CSV file can be populated with information for the catalog from other sources, such as databases or text files. You can create your input files in a Microsoft Excel spreadsheet, use Excel to save the file in CSV format and import the data into your CA Service Catalog catalog.

The IXUTIL utility reads, parses and temporarily stores data in XML format then re-reads the XML file and loads the database with data from that file.

Note: The values in a CSV file cannot contain semi-colons (;), carriage returns, or any other characters prohibited by CSV format. Therefore if you want to import a CSV file, before creating or modifying the file in Excel or any other application, review the application's rules for CSV format, including prohibited characters. While working in the application, save the files frequently in CSV format to help detect any prohibited characters. If you attempt to save a file with a prohibited character in CSV format, the application should display a warning, and you should specify a different character.

The ixutil.cfg configuration file describes the mapping of columns in the input file to the attributes of catalog objects. This mapping is defined in the “#CSV property mappings” section of the ixutil.cfg file. All supported object and property names are given in the ixutil.cfg file. Only the fields in the CSV file that have matching entries in the ixutil.cfg file will be imported. This means there can be extra entries in the ixutil.cfg file or extra fields in the CSV file without causing an error.

The ixutil.cfg file also describes any default values to be set for the attributes of catalog objects when values are missing from the CSV file. These default values are set in the “#default fields” section of the ixutil.cfg file.

The first row in the CSV file defines the column names that are mapped to real properties as defined in ixutil.cfg. When using Excel, these column names come from the Excel spreadsheet column header values prior to saving the spreadsheet in CSV format and may be set to any value that makes sense to the user. There are four types of catalog objects that can be populated:

- Folder
- Service
- Service option group
- Service option element

Folders can contain other folders, so each row in the Excel spreadsheet can have multiple “folder” columns representing the hierarchical folder structure.

For example, if the highest number of hierarchical folders your catalog has is two, your spreadsheet would use two folder-related columns, perhaps using heading values of Folder1 and Folder2.

In addition, service option groups contain service options which contain service option elements. Each row in your spreadsheet contains the definition of one service option element with the other values for the folder hierarchy and service repeated or left blank. If a value is left blank, the value from the preceding row is used.

For example, suppose you want your catalog to have two desktop-related services (Standard Desktop and Deluxe Desktop) to be presented in a folder hierarchy (*Desktops* under *IT Hardware*). Note that there are two service option elements (a name and a price) for each service option and two service options (the hardware and some software) for each service.

IT Hardware

Desktops

Standard Desktop	2 GHz processor, 100 GB drive	\$699.00/unit
	Productivity Software Pack	\$100.00/unit
Deluxe Desktop	3 GHz processor, 200 GB drive	\$799.00/unit
	Productivity Software Pack Pro	\$150.00/unit

In this case, your spreadsheet would include this data (as well as other columns):

Folder1	Folder2	Service_Name	SOGElement_Text
IT Hardware			
	Desktops		
		Standard Desktop	2 GHz processor, 100 GB hard drive \$699.00/unit
			Productivity Software Pack \$100.00/unit
		Deluxe Desktop	3 GHz processor, 200 GB hard drive \$799.00/unit
			Productivity Software Pack Pro \$150.00/unit

Syntax of Property Mappings

CSV_field_name=object_name.field_name

where:

CSV_field_name

Matches the field name in the CSV file being imported (the column header in the source Excel spreadsheet) and can be any unique value set by the user.

object_name

Is one of these values: folder, service, serviceoptiongroup or serviceoptionelement.

field_name

Is one of the valid database field names supported for each object. The complete list of field_name values for each object is listed in the provided ixutil.cfg file.

Example

Service_Name=service.offering_name
SOGElement_Text=serviceoptionelement.item_text

where:

Service_Name

Is the column name in the CSV file being imported that maps to the service.offering_name data element in the catalog (the service name).

SOGElement_text

Is the column name in the CSV file being imported that maps to the serviceoptionelement.item_text data element in the catalog (the service option element text value).

You can also define default values in the “#default fields” section of the configuration file. The default values are used if a value for the object_name.field_name is not specified in the CSV file.

If properties are defined in both the mappings and defaults sections then default field entry is ignored.

Syntax of Default Fields

defaults.object_name.field_name=value

Example:

defaults.offering.version=11.0
defaults.offering.date_created=05/18/2006 5:0:0

To import catalog entries from a CSV file

1. First verify your mapping definitions in ixutil.cfg are correct (every needed field in the file being imported is mapped to a catalog attribute).
2. Open a command prompt window and navigate to %USM_HOME%\scripts folder.
3. Type in the command:


```
ixutil import offeringrateplan_csv -f
c:\mypath\sample.csv
-domain mybusinessunit
```

Where "mybusinessunit" is the business unit you want your catalog data to be imported to and c:\mypath\sample.csv is the name of the CSV file to be imported.

The services defined in the CSV file are added to the catalog definition and can be viewed using the Service Builder menu option.

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.

To import CA Workflow forms

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 Service View computer, open a command prompt and enter the following command:

```
ixutil import form -f {filename\foldername} -format xml -cawf -domain 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.

-domain 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 `-domain subBud`.

When you use the `-domain domain-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 `"-domain ABC,"` then the imported service named Diamonds overwrites the existing service named Diamonds (if one exists) in the domain named ABC.

-f {*filename* | *foldername*}

Imports either an individual CA Workflow form or all CA Workflow forms from a specified folder, as follows:

To import an individual CA Workflow form, specify `-f filename`, where *filename* specifies the complete path name of the form.

To import all the CA Workflow forms from a specific folder, use `-f foldername`, where *foldername* specifies the complete path name of the folder.

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.

Import and Export 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 domains, all forms in a folder, or a specific form.

Note: For more information about the `ixutil` command, including other syntax options not listed here, see the *Reference Guide*.

1. On the source Service View computer, export the form or forms by opening a command prompt and entering the following command:

```
ixutil export form -f filename -format xml {-folder foldername} -form formname} -domain id
```

-f filename

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.

-domain 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 domains.

For example, if the business unit login ID is `subBud`, specify `-domain 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 filename` parameter described earlier.

Sample commands follow:

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

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

2. Copy the exported XML file to the target Service View computer.

3. On the target Service View 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 filename -format xml -domain id
```

-f *filename*

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.

-domain *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 `-domain 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 "`-domain ABC`," then the imported service named Diamonds overwrites the existing service named Diamonds (if one exists) in the domain 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.
- If a form or folder being imported *does* already exist on the target computer, the system compares the last date modified of the duplicate objects and does the following:
 - If form or folder on the source computer is older than the one on the target machine, no update occurs.
 - If form or folder on the source computer is newer than the one on the target machine, the system overwrites the existing form on the target computer with the newer one from the source computer.

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).

4. Verify that the forms appear correctly in the Form Designer; update them if necessary.
5. Test the newly imported services and their forms.
6. Make the services available for users.

Import and Export Form Designer Forms Associated to Service Option Elements

You can export and import Form Designer forms to include them when exporting and importing 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 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.

1. On the source Service View 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 filename -format xml {-folder name|-form name} -domain id
```

```
ixutil import form -f filename -format xml -domain id
```

For an explanation of the command options, see the descriptions of these commands in [Import and Export Form Designer Forms](#) (see page 37).

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

```
ixutil export {offeringrateplan|rateplan} -f filename {-offering name|-folder name|-domain id} -include_forms other options
```

offeringrateplan

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

rateplan

includes service option groups (rate plans) and service option elements (rate items).

Note: Service option elements are processed in `ixutil export offeringrateplan` and `rateplan` commands *only*, *not* in `ixutil export offering` commands. Therefore, use `--include_forms` in `ixutil export offeringrateplan` and `rateplan` 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|-domain id}

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

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 *filename* parameter.

3. Copy the exported XML file to the target Service View computer.
4. On the target Service View computer, open a command prompt and enter the following command to import the forms:

```
ixutil import {offeringrateplan|rateplan} -f filename other options
```

If you specified *offeringrateplan* in the original export command, specify *offeringrateplan* in the corresponding import command. Conversely, If you specified *rateplan* in the original export command, specify *rateplan* in the corresponding import command.

-f filename

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 services and their forms.
7. Make the services available for 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 migrate new and updated objects, including policies, from one computer to another. If necessary, you export and import policies when you upgrade or migrate CA Service Catalog. You also export and import policies when you upgrade your computer or if a computer fails.

To export and import policies

1. Open the CA Service Catalog command prompt on any Service View 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 filename
```

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

```
ixutil export policy -f filename - domain business unit name
```

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

```
ixutil export policy -f filename -policy name -domain name
```

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 filename -folder name -domain name
```

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

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

```
ixutil import policy -f filename -domain name
```

The `-domain business unit name` 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 filename -domain business unit name -disable
```

The following options may require explanation:

-f *filename*

Specifies the name of the file being exported or imported. This file contains the policy object or objects that you are exporting or importing.

The Catalog system uses an XML file to export and import policy objects.

-domain *name*

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 `-domain` 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 `-domain` 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.

-disable

Imports the policy or policies in disabled status.

By default, policies 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.

3. Log in to CA Service Catalog on the computer where you imported the policies. If applicable, log in to the business unit to which you exported the policy or policies.
4. Click Service Builder, Policy Builder.
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.

IXCATALOG Import/Export Utility

IXCATALOG is an importing and exporting command line utility that preserves and migrates catalog entries.

The catalog entry data is exported into several files using proprietary XML format and the same XML format must be used in the like named data files when importing. You can determine the correct XML format by examining an exported XML file. The exported files are named as follows:

- `sla_offering.xml`
- `rateplan.xml`
- `offering.xml`

IXCATALOG Syntax

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

```
ixcatalog export [filenameprefix]
```

filenameprefix

Specifies an optional parameter used to prefix the output files.

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

```
ixcatalog import [filenameprefix]
```

filenameprefix

Specifies an optional parameter used as a prefix to identify the input files.

For Windows, the `ixcatalog.bat` file is located in the `%USM_HOME%\catalog\scripts` folder.

IXCATALOG is configured from 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 IXCATALOG appear in the `%USM_HOME%\logs\ixcatalog.log` file.

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.

Appendix A: Java Plug-Ins

This section contains the following topics:

[Java Report Plug-Ins](#) (see page 47)

[Java Rule Action Plug-Ins](#) (see page 55)

[Java Scheduler Plug-Ins](#) (see page 56)

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 <code>com.ca.usm.reporting.Plugins</code>)	Input Parameters (Prompt) / [ConstantValue]	Output Variables
CA Service Accounting Folder			
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:Stri ng (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, pvar
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
<p>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.)</p>	<p>RequestFulfillmentReport</p>	<p>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]</p>	<p>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</p>

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)	SMEEventFunc	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](#) (see page 57)

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 CA Service Accounting 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 CA Service Accounting 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 CA Service Accounting 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 CA Service Accounting 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 CA Service Accounting 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 CA Service Accounting 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 CA Service Accounting 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_processes\$	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_subclasses\$	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 CA Service Accounting where: 0=credit 1=charge	1
\$charge_date\$	Date the item will first be charged for by CA Service Accounting	
\$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 CA Service Accounting	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 CA Service Accounting	
\$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 CA Service Accounting. 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\$
