CA Service Catalog

Administration Guide

Release 12.7.00

technologies

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 3: Managing Users and Roles   77

Chapter 4: Managing Reports Using the Report Builder   97

Chapter 5: Managing Dashboards   117

Chapter 6: Using Web Services   123

Chapter 7: Using Service Builder   139

Chapter 8: Using the Form Designer 277

Chapter 9: API Plug-ins   381

Chapter 10: Using CA Service Accounting   387

Chapter 11: Data Mediation   427

Chapter 12: Using Policies to Manage Requests 451

Chapter 13: Using CA Service Catalog for Administrators 507

Chapter 14: Using CA Service Catalog for Catalog Users 561

Glossary 639

Chapter 1: Administration Functions and Tools

This section contains the following topics:

Overview

Administrators have access to additional administrative functions and tools not available to other user roles. Administrators can perform the following tasks, grouped according to the main tabs on the user interface:

- Home

  - Manages dashboards

  - Manages messages

  - Manages reports

  - Manages documents

  - Makes requests for other users or accounts

  - Manages requests for users and accounts within the scope of the administrator

  - Configures applications

- Service Builder

  - Manages the services catalog

  - Manages service option groups

  - Manages configuration settings for CA Service Catalog

- Form Designer

  Creates and maintains forms used in requests

- Policy Builder

    Creates and maintains policies for automating the assignment of requests pending action to users and groups

- Accounting

    – Manages accounts

    – Manages invoices

    – Manages budgets

    – Manages adjustments

    – Manages configuration settings for CA Service Accounting

- Administration

    – Manages business units and accounts within the scope of the administrator

    – Manages users within the scope of the administrator

    – Manages and build reports

    – Configures and performs data imports using data mediation

    – Accesses the administration tools such as events, rules and actions, the scheduler, and other tools.

    – Manages configuration settings for CA Service Catalog

- Administration Quick Start dashboard

    Users with the Service Delivery Administrator role automatically have access to this dashboard. It contains links to commonly used administration functions.

Note: For more information about configuration, see the *Implementation Guide*.

Messages and Alerts

Change events are listed on the Change Events page of the Home, Messages tab. The Catalog system generates change events automatically when either the system or users (especially administrators) perform significant actions. For example, if you add a user (see page 81) or edit a user (see page 86), your action appears as a change event message on the Change Events page.

Alerts are listed on the Alerts page of the Home, Messages tab. Typically, an alert is issued when a system action fails or a user action fails. Thus, alerts represent failed change events. The Alerts page lists failed events issued by the Scheduler (see page 41), Service Builder (see page 140), Invoice Engine (see page 394), and other components of CA Service Catalog. For example, failed actions and alerts can result from an administrator specifying an incorrect email address when creating a user. In such cases, actions that use the email address can fail: An attempt by a request manager to perform email-based approval of a request from that user can fail, resulting in the "REQEMAILACTION_FAILED" alert.

Administrators can use the following functions on the Messages menu to manage messages and alerts:

- Manage News Messages

  - Enables users to add a news message (see page 20) for other users or roles in the same business unit.

  - Enables users to delete their own news messages.

  - Enables Service Delivery Administrators to delete any news messages from users.

- Manage Change Events

  - Identifies details about the object affected by events, for Service Delivery Administrators and Super Business Unit Administrators.

  - Enables Service Delivery Administrators to view (see page 20) and delete change event messages.

- Manage Alerts

  - Indicates system alerts for Service Delivery Administrators and Super Business Unit Administrators.

  - Enables Service Delivery Administrators to view (see page 21) and delete system alerts.

  - Enables Service Delivery Administrators to retry failed rule actions (see page 21) specified by CA Workflow process definitions or CA Process Automation processes.

Add a News Message

You can add a news message to the News Messages list.

Follow these steps:

1. Select Home, Messages, and select News from the Messages menu.

    The News Messages list appears.

    Note: A Service Delivery Administrator can also delete news messages from this list.

2. Click Add News Message.

    The Add News Message page appears.

3. Complete the fields to meet your requirements and click OK.

    The news message is saved and the News Messages list appears. The list includes the newly added message.

You have added a news message.

View Change Events

You can view change event messages from the Messages tab.

Follow these steps:

1. Select Home, Messages, and select Change Events from the Messages menu.

    The Change Events Messages list appears. By default, only the change event messages for the current day for the current business unit appear.

2. (Optional) Do any or all of the following:

    ■   Change the business unit whose events appear on the page.

    ■   Filter the events, according to your selection criteria. A Service Delivery Administrator can delete one or more change event messages from this page.

    ■   View details about individual change events.

You have viewed the change event messages.

View Alerts

You can view alert messages.

Follow these steps:

1. Select Home, Messages, and select Alerts from the Messages menu.

    The Alert Messages list appears. By default, only the alert messages for the current day for the business unit of the logged in user appear.

2. (Optional) Do any or all of the following:

    ■   Change the business unit whose events appear on the page.

    ■   Filter the events, according to your selection criteria. A Service Delivery administrator can also delete alert messages and display alerts for failed actions. Such actions failed to start a CA Workflow process definition or a CA Process Automation process.

    ■   View details about individual alerts. The detailed information can sometimes represent internal ID values.

You have viewed the alert messages.

Retry Failed Actions

When a rule action fails to launch its associated CA Workflow process definition or a CA Process Automation process, a system alert is created. Possible causes for the failure include the related service either not running or being too busy to accept more work. Service Delivery administrators can retry failed actions of these types.

Follow these steps:

1. Select Home, Messages, and select Alerts from the Messages menu.

    The Alert Messages list for all alert types appears.

2. Click List All Failed Actions.

    The Alert Messages list changes to display the alerts of type WFACTION_FAILED and ITPACTION_FAILED.

3. Use Filter Alerts and Change Business Unit to display the list of alerts that includes the failed actions you want to retry.

4. Select the alerts related to the failed actions that you want to retry and click Retry Failed Actions. When prompted, confirm your action.

    The Alert Messages list refreshes to reflect your retry attempt.

You have retried the failed actions.

Documents

CA Service Catalog enables users to view shared documents. Documents are organized into a folder structure in the Documents tree. Service Delivery Administrators, Super Business Unit Administrators, Business Unit Administrators and End Users can manage documents and folders. Administrators grant access to folders and documents according to role.

You can perform the following functions for documents:

- View a document folder (see page 22)

- View a document (see page 23)

- Add a document folder (see page 23)

- Upload a document (see page 24)

- Copy and paste documents and folders in the Documents tree

- Delete documents and folders

View a Folder

You can view a folder in the Documents tree so that you can access the documents in the folder.

To view a folder

1. Click Home, Documents.

   The Documents page appears.

2. Select the Documents tree folder you want to view.

3. Select the Properties tab.

   You can view and edit the properties of folder.

4. Select the View Content tab.

   You can view the contents of the folder.

View a Document

You can view a document in the Documents tree folder.

To view a document

1. Click Home, Documents.

    The Documents page appears.

2. Select the Documents tree folder you want to view.

3. Select the Properties tab.

    You can view and edit the document properties.

4. Select the View Content tab.

    You can view the document.

Add a Document Folder

You can add folders to the Documents tree to help organize documents efficiently.

To add a document folder

1. Click Home, Documents.

    The Documents page appears.

2. Select the Documents tree folder under which you want to add a new folder.

3. Select New Folder from the Action drop-down list and click Go.

    The Add New Folder window appears.

4. Enter the property values for the new folder.

    Note: Click the help icon for more information about the properties.

5. Click OK.

    The properties of the new folder are saved, and the new folder is added to the Documents tree.

You have added the new folder to the document tree.

Upload a Document

You can upload a document to the Documents tree folder so that other users can access it.

To upload a document

1. Click Home, Documents.

    The Documents page appears.

2. Select the Documents tree folder under which you want to add a new document.

3. Select Upload Document from the Action drop-down list and click Go.

    The Add New Document window appears.

4. Enter the property values for the new document.

    Note: Click the help icon for more information about the properties.

5. Click OK to save the properties of the new document and upload the document.

You have added the new document to the Documents tree.

Search

You can search for the following objects in CA Service Catalog:

■ Accounts (see page 74)

■ Invoices, if CA Service Accounting is installed

■ Payments, if CA Service Accounting is installed

■ Requests (see page 587)

The search criteria can be simple (account name) or advanced (multiple criteria). You can save queries for reuse later.

Administration Tools

Using the Administration, Tools menu, administrators can select from the following options and perform the associated actions:

- Events, Rules, and Actions (see page 26)

  Manage events and the rules and actions associated with them

- Scheduler (see page 41)

  Manages scheduled tasks

- Links

  Provides access to the following:

    - The CA Workflow Process Definition Tool (IDE), Worklist, or Process Manager

    - The CA EEM user interface

    - CA Service Catalog web services, including its API documentation

    - CA Business Intelligence InfoViewWeb, which enables you to create and run BusinessObjects Enterprise reports

    - The CA Service Catalog documentation bookshelf

Events, Rules, and Actions

Events represent changes that occur within CA Service Catalog. Several *standard* events typically occur in various components. You can also add *custom* events.

Events can have *rules* associated with them. Rules can have a set of filter conditions that define when the rule applies. When the filter conditions are satisfied and the rule is enabled, the rule *actions* are launched.

Each standard event occurs when an action in CA Service Catalog causes it. For example, the User Create event occurs when an administrator adds new user, by using either the user interface or the createUser web service method.

A *custom* event occurs when you do one of the following:

- Use one of the postEvent web service methods for the event type
- Use a URL to post the event

The Events-Rules-Actions menu displays the event types and provides access to the rules and actions associated with each event type. You can use these menu options to do the following:

- Add a custom event type (see page 26)
- Post a logical event (see page 38)
- Manage the rules for an event type (see page 27)
- Manage the actions for a rule (see page 32)

Important! For specific information about using the built-in rules and actions in the CA Workflow simple or complex approval or fulfillment processes, see the *Integration Guide*.

Add a Custom Event Type

You can optionally create *custom* events (see page 26) by adding a new event type. You typically do so to address a need in your organization that the predefined event types do not meet.

To add a new event type

1. Select Administration, Tools, Events-Rules-Actions.

   The Event Types page appears.

2. Click Add.

   The Event Type Information page appears.

3. Complete the fields on the page. The following fields are not self-explanatory:

    Note: The event type name is not required to be unique. However, we recommend that you specify a unique name for usability.

    Event Source

    Specifies that the event source is *physical* or *logical*.

    A physical event is based on an update to a database table and is for standard events only.

    Audit Trail Level

    Specifies the level of detail to log in an event log table.

    Transaction Name

    For a physical event, specifies the name of the database table.

    For a logical event, specifies any name.

    Transaction Type

    Specifies the transaction type: Added, Modified, or Deleted.

    This type represents the general behavior of the custom event. The Transaction Type together with the Event Type Name and Event Source make a unique combination.

4. Click OK.

    The event is added to the Event Types list.

Note: You can specify event variables in filters and actions in rules. For a complete list of event variables, see the appendix "Event Variables" in the *Reference Guide*. The descriptions in that appendix for request change events, request item form element change events, and request/subscription item change events can be especially helpful.

Management of Rules

You can configure rules in several ways, to trigger the actions you want under the conditions you want. For example, you can create and configure a rule to trigger an action when a user, service, business unit, or account is created or updated.

The main activities for managing rules follow:

- Add a rule (see page 28)
- Review event filters (see page 28)
- Add an event filter (see page 29)
- Enable or disable a rule (see page 30)
- Delete a rule (see page 30)

Add a Rule

You can add a new rule (see page 26) for several reasons. For example, you can create and configure a rule to trigger an action, such as an email notification, when an approver approves or rejects a request pending action.

To add a rule

1. Select Administration, Tools, Events-Rules-Actions.

   The Event Types page appears.

2. Click the event to which you want to add the rule.

   The Event Type Details page appears.

3. Click Add.

   The Add Rule page appears

4. Complete the fields, following these guidelines:

   ■ Specify a unique rule name.

   ■ Click the Add Filter icon to create or edit an event filter (see page 28).

5. Click OK to save the rule definition.

   The rule is saved. The Event Type Details page appears, and it includes the new rule.

Event Filters

When adding or updating a rule (see page 28), administrators can add an event filter (see page 29) to the rule to specify when the rule applies to an event. The rule triggers associated actions *only* if the conditions specified in the event properties are met.

The properties available to a rule condition depend on the event type of the rule. You can specify both *old* and *new* (or *before* and *after*) values for each event property in the condition.

Without an event filter, the rule is always launched and the associated actions are always executed if the event occurs.

Add an Event Filter

You can add an <u>event filter</u> (see page 28) to a <u>rule</u> (see page 26) to specify the conditions for invoking the rule and triggering its actions. Otherwise, the rule and actions are invoked whenever the event occurs.

To add an event filter to a rule

1. Select Administration, Tools, Events-Rules-Actions.

   The Event Types page appears.

2. Click the event that contains the rule of interest.

   The Event Type Details page appears.

3. Click the Edit icon for the *custom* rule to which you want to add the event filter.

   Note: You can add or update event filters in custom rules *only*, not predefined (built-in) rules. Predefined rules permit very limited editing. If necessary, click the Copy icon to copy a predefined rule and save it with a new name as a custom rule.

4. Click the Add icon next to the Event Filter field

   The Create Event Filter dialog appears.

5. Specify values for the property, operator, and constant of your new condition in the Condition Builder section of the dialog.

6. Click the Add Condition arrow to add your new condition to the Current Conditions section.

   The Condition appears in the Current Conditions field. It is translated into a conditional SQL-style expression that the rule engine can process.

7. (Optional) Add another condition, if required.

   You can add as many conditions as required.

   Note: You can delete an existing condition by selecting that row and clicking the Delete trash can icon.

8. Click OK when you are finished.

   The Event Filter appears in the Rule Information section of the Edit Rule page for the selected rule.

9. Click OK.

   The system saves the rule, including the event filter.

You have added the event filter.

Note: You can specify event variables in filters and actions in rules. For a complete list of event variables, see the appendix "Event Variables" in the *Reference Guide*. The descriptions in that appendix for request change events, request item form element change events, and request/subscription item change events can be especially helpful.

Enable or Disable a Rule

You can enable a rule (see page 26) when your organization is ready to use it. Conversely, you can disable a rule when your organization no longer needs or wants to use it.

To enable or disable a rule

1. Select the box next to the rules to be enabled or disabled in the Rules list from the Event Type Details page.

2. Click Enable or Disable.

   This changes the status of each selected rule.

Delete a Rule

To delete a rule

1. Select the box next to the rules in the Rules list to be deleted from the Event Type Details page.

2. Click Delete.

   Each selected rule is deleted and removed from the Rules list.

Actions

A rule can have one of more actions associated with it. When an event occurs and a rule filter is satisfied, the associated rule actions are executed, if the rule is enabled.

A rule action causes some task to be executed. A rule action can perform one of several action types such as: send an email, start a CA Workflow process definition, run a provided Java plug-in, and so on. The properties available for each event type are set to values appropriate to the event context. These properties can be treated as *variables* and included in actions. For example, an action can send an email which included the property values in the message body or an action can start a CA Workflow process definition or a CA Process Automation process and pass parameters set to the property values.

Actions can be of various types, such as:

Command Line

Run a command on the server.

HTTP Post

Execute an HTTP Post to a URL.

Java

Execute a provided Java plug-in.

Email

Send an email.

CA IT PAM

Start the specified CA Process Automation process, passing any necessary parameter values.

CA Workflow

Start the specified CA Workflow process definition, passing any necessary parameter values.

Reservation Manager

Specifies an action for use with Reservation Manager.

For details about the integration between CA Service Catalog and Reservation Manager, see the *Integration Guide*.

Note: You can specify event variables in filters and actions in rules. For a complete list of event variables, see the appendix "Event Variables" in the *Reference Guide*. The descriptions in that appendix for request change events, request item form element change events, and request/subscription item change events can be especially helpful.

Management of Actions

You can configure actions in several ways, to trigger the response you want for specific rules. For example, you can create and configure a rule to trigger an automatic email notification when a fulfiller fulfills a request.

The main activities for managing actions follow:

- Add an action to a rule (see page 32)

- Disable an action (see page 36)

- Enable an action (see page 37)

- Delete an action (see page 37)

Add an Action

You can add a new action (see page 31) to a rule (see page 26) to meet a need in your organization that the predefined actions and existing custom actions do not meet. For example, you can add one new action to run a command when a request of a certain class changes status from Pending Approval to Approved.

Note: You can *add* new actions (create custom actions) in both custom rules and predefined (built-in) rules.

To add an action to a rule

1. Select Administration, Tools, Events-Rules-Actions.

   The Event Types page appears.

2. Click the event that contains the rule and action of interest.

   The Event Type Details page appears.

3. Click the Edit icon for the rule to which you want to add the action.

   The Rule Details page appears.

4. Click Add.

   The Add Action page appears.

5. Complete the parameters (see page 33) on that page.

6. Click OK to save your changes.

You have added the action.

Edit an Action

You can edit a custom action (see page 31) in a rule (see page 26) when required to meet a need in your organization that the predefined actions or custom actions do not meet. You can perform significant edits on custom actions *only*. Predefined actions permit only limited editing. When required, you can edit a custom action to meet a need in your organization more closely. For example, you can edit a custom action to refine the request email that the action sends.

To edit an action

1. Select Administration, Tools, Events-Rules-Actions.

   The Event Types page appears.

2. Click the event that contains the rule and custom action of interest.

   The Event Type Details page appears.

3. Click the Edit icon for the rule that contains the action that you want to add.

   The Rule Details page appears.

4. Click the Edit icon for the action you want.

   The Edit Action page appears.

5. Complete the parameters (see page 33) that are open for editing.

6. Click OK to save the changes.

You have edited the action.

Parameters for an Action

You specify the several parameters for a custom action when you add (see page 32) or edit (see page 33) the action. The following parameters are not self-explanatory:

Note: After you create, save, and close a custom action, you can edit its parameters, except that you *cannot* change the Type field. To specify a different type for the action, copy and rename it. If you do so, update the related rule to trigger the renamed action.

Type

Specifies the type of action to use. To disable a rule action, select DISABLED. Depending on the type you select, additional fields are exposed. The add icon can be used to add an event parameter to the field used for each action type.

Command line

Specifies a command to run.

HTTP Post

Specifies the URL to post.

Java

Specifies the Java Class and parameters to pass.

Note: For more information about available Java classes, see the *Reference Guide*.

Email

Notifies users (typically administrators) about system activities not related to requests. Examples include the addition of new users or a new business unit, or a data mediation load.

Request Email

Notifies stakeholders about a request and optionally includes request details.

The following parameters require explanation:

Include Request Details - Specifies whether to add the details of the request to the text of the message body in the request email.

Specify Yes to add these details or No to omit them.

Type – Applies *only* when you use the $pending_action_users_or_groups$ variable.

Specifies the Type: Request, Service, or Service Option.

This parameter retrieves the list of pending action users or groups applicable to the selected Type. This parameter also assigns the retrieved list to the $pending_action_users_or_groups$ variable.

Request ID – Specifies the ID (not the name) of the request, which must be a numeric value or variable.

Request Item ID – Applies only when the selected Type is Service or Service Option.

Specifies the ID of the subscription detail object, which must be a non-numeric value or variable. This value is used to retrieve the list of pending action users or groups applicable to the selected Type (Service or Service Option).

From Email – Specifies the email address from which to send request emails.

From Name – Specifies the name of the email address from which to send request emails.

Note: If you do not specify a value for From Email or From Name, the Catalog system assigns the following value: the value specified in the Service Builder, Request Management configuration options of the business unit of the request.

To – Specifies one or more primary recipients for the email.

You can use specific email addresses, user IDs, and requested for user-related variables.

To send an email to specific users or groups automatically, specify the matching variables, as follows:

- For the requested for user, specify $req_for_user_id$

- For the requested by user, specify $req_by_user_id$

- For the pending action user, specify $pending_action_users_or_groups$

To specify more than one value, separate the entries with a semi-colon (;).

You can select these variables and other variables from the list that appears when you click the list icon next to the field.

CC – Specifies one or more CC recipients for the email.

Use the same syntax as the To field.

BCC – Specifies one or more BCC recipients for the email.

Use the same syntax as the To field.

Subject – Specifies the text for the subject of the email.

Message Body – Specifies the text for the message body of the email.

Note: A commonly used variable is $server_url$. You can optionally use it to direct the recipient to the request detail page.

CA Process Automation or CA Workflow

Specifies the start request form (SRF) (for CA Process Automation) or process definition (for CA Workflow) to invoke.

Use the Find Start Request Form icon (for CA Process Automation) or Find Process Definitions icon (for CA Workflow).

Specify the parameter values for the SRF or process definition, if applicable.

Reservation Manager

Specifies a parameter to pass to Reservation Manager.

For details about this integration, see the *Integration Guide*.

Execution Mode

Specifies whether the action runs asynchronously or synchronously.

Note: Administrators *cannot* set the order in which actions run.

Timeout

Specifies the number of seconds the action must run before a user can cancel it.

A value of 0 means no timeout applies.

Multiple Values

For some events, some parameters can have multiple values. For these parameters, use a delimiter to separate multiple values. For each parameter that can have multiple values, you can specify the separator that is used when that event occurs. If the Multiple Values check box is selected, the rule action is executed for each value for the parameter indicated in the Parameter field.

Note: The only standard event with multiple values parameters is the Catalog Subscription Change event. This event provides multiple values for the $new_offerings$, $new_subscriptions$, $old_offerings$ and $old_subscriptions$ parameters. You can use multiple value parameters with custom events; However, the event posting logic must provide the multiple parameter values.

Disable an Action

You can disable both custom and predefined actions (see page 31) in rules (see page 26). For example, in the same rule, you can enable (see page 37) one action and disable another action in response to a change in requirements or policy. Similarly, if you do not want to use the action that is enabled by default, you can disable it.

To disable an action

1. Select Administration, Tools, Events-Rules-Actions.

   The Event Types page appears.

2. Click the event that contains the rule and action of interest.

   The Event Type Details page appears.

3. Click the Edit icon for the rule that contains the action that you want to disable.

   The Rule Details page appears.

4. Select the action you want to disable in the Actions list.

5. Click Disabled.

   The action is disabled.

You have disabled the action.

Enable an Action

You can enable actions (see page 31) in rules (see page 26) when you are ready to use the actions. You can enable both custom and predefined actions. For example, in the same rule, you can enable one action and disable (see page 36) another action in response to a change in requirements or policy. Similarly, if you want to use an action that is disabled by default, you can enable it.

To enable an action

1. Select Administration, Tools, Events-Rules-Actions.

    The Event Types page appears.

2. Click the event that contains the rule and action of interest.

    The Event Type Details page appears.

3. Click the Edit icon for the rule that contains the action that you want to enable.

    The Rule Details page appears.

4. Select the action you want to enable in the Actions list.

5. Click Enabled.

You have enabled the action.

Delete an Action

When editing rules (see page 26), you can delete custom actions (see page 31) *only*. You *cannot* delete predefined (built-in) actions. Delete custom actions *only* if you are certain that your organization does not need them. For example, you can delete an obsolete an action that you have replaced with a newer one.

To disable an action

1. Select Administration, Tools, Events-Rules-Actions.

    The Event Types page appears.

2. Click the event that contains the rule and action of interest.

    The Event Type Details page appears.

3. Click the Edit icon for the rule that contains the action that you want to disable.

    The Rule Details page appears.

4. Select the action you want to delete in the Actions list.

5. Click Delete.

The action is deleted.

Post a Logical Event

You can post a logical event either through HTTP Post or from one of the postEvent Administration web service methods.

To specify an HTTP URL to post an event to the system, use the following syntax:

http://hostname:port/usm/wpf?Node=icguinode.postevent&username=userid&pass=password&domain=business unit&Args=eventsource&Args=nsppath&Args=transactionname&Args=eventtypename&Args=transactiontype&Args=eventdescription&Args=associatedobjectid&Args=false&Args=param1|oldvalue1!param#|oldvalue#!&Args=param1|newvalue1!param#|newvalue#!

The following parameters require explanation:

userid

> Specifies a valid user ID for authentication.

password

> Specifies the password for the user ID.

businessunit

> Specifies the business unit for the role of the user ID.

eventsource

> Specifies whether the source is logical or physical.

nsppath

> Specifies the namespace path, a placeholder value only. The Catalog system does not use the actual value but requires a placeholder value.

transactionname

> Specifies the transaction name for the event.

eventtypename

> Specifies the name of the event type.

transactiontype

> Specifies the transaction type for the event.

eventdescription

> (Optional) Specifies a description for the event.

associatedobjectid

> Specifies the ID of an object to associate with this event.

> You can optionally specify one of the event parameters for this value.

param#|oldvalue#

Specifies the parameter name and the *old* value. Delimit the name and value with a vertical bar. Separate each name and value pair with an exclamation point.

param#|newvalue#

Specifies the parameter name and *new* value. Delimit the name and value with a vertical bar. Separate each name and value pair with an exclamation point.

Note: The Catalog system uses old and new values when evaluating any rule filters associated with event rules.

Example: Post a Custom Logical Event

The following sample URL posts a custom logical event to the Service View server on Server B:

http://serverB:8080/usm/wpf?Node=icguinode.postevent&username=spadmin&pass=spadmin&domain=ca.com &Args=LOGICAL&Args=nsppath&Args=StorageLimitHandler&Args=EXCEEDED_STORAGE_LIMIT&Args=LO GICAL&Args=Storagelimitexceeded&Args=uid&Args=false&Args=storageLimit|10!&Args=storageLimit|20!

The event for this post contains the following name and value pairs:

username=spadmin

Specifies the name of user posting this event.

pass=spadmin

Specifies the password of this user.

domain=ca.com

Specifies the name of domain in question.

Args=LOGICAL

Specifies the type of event.

Args=nsppath

Specifies the namespace path (any value).

Args=StorageLimitHandler

Specifies the transaction name.

Args=EXCEEDED_STORAGE_LIMIT

Specifies the transaction event type.

Args=LOGICAL

Specifies the transaction type.

Args=Storagelimitexceeded

Specifies the description.

Args=uid

Specifies the object ID of this event. The uid event parameter is used here.

Args=false

Indicates whether this event is a partial event. The value is always *false*.

Args=storageLimit|10!

Specifies the parameter name and the old value.

Args=storageLimit|20!

Specifies the parameter name and the new value.

Note: You can enter such a URL as an action for a rule of an event, or from a URL from a Java program. In both cases, replace the symbol *&* with *%26*.

More information:

Using Web Services (see page 123)

Scheduler

You can use the Scheduler to schedule in advance a task that you want to run one or multiple times. Scheduling is especially useful for recurring tasks, so that you do not have to perform them manually. Use the Scheduled Tasks list of the Scheduler tool to manage scheduled tasks, as follows:

- View the list of scheduled tasks

- Add a scheduled task (see page 41)

- Edit a scheduled task (see page 43)

- Delete a scheduled task (see page 43)

- View Scheduler-related system alerts

Note: For optimum scheduling capabilities, use CA Process Automation to schedule tasks. For details, see the CA Process Automation documentation.

Add a Scheduled Task

You can use the Scheduler (see page 41) to schedule a new task that you want to run one or multiple times. Adding a scheduled task is useful to help implement new or updated procedures in your organization.

Note: For optimum scheduling capabilities, use CA Process Automation to schedule tasks. For details, see the CA Process Automation documentation.

To add a scheduled task

1. Click Administration, Tools, Scheduler.

   The Scheduled Tasks list appears.

2. Click Add.

   The Create Scheduled Task window appears.

3. Complete the fields.

The following options for the *Category* field require explanation:

Valid From

Specifies the date when the scheduled task starts running.

Valid Until

Specifies the date when the scheduled task stops running. If this field left blank, the scheduled task continues to execute with no end date.

Recur

Specifies the recurrence interval for running this task. The Recur setting together with the Rule settings determine when the task executes.

Rule

Specifies the rule settings for running this task.

Action Type

Specifies *one* of the following. Depending on the Action Type, additional fields appear to help further define the action.

Execute Command Line – Specifies a command to run on the server.

Execute Offline Data View – Creates an offline data view based on a data object, as follows:

■   *Choose Report* selects a data object from a list.

■   *Set Options* specifies runtime variable values and sets other offline report options.

■   *Offline Reports* selects an offline report from a list or specifies a new file name.

Execute Offline Layout – Creates an offline layout based on a data view.

■   *Choose Report* selects a data view from a list.

■   *Set Options* specifies runtime variable values and sets other offline report options.

■   *Offline Reports* selects an offline report file from a list or specifies a new file name.

Execute Scheduler Plugin – Specifies an option for system use *only*.

Note: If you have scheduled data mediation tasks, you can view this option.

Create a CA Workflow Process Instance – Runs the specified CA Workflow process definition and passes any required parameter values.

You can use the Find Process Definitions magnifying glass icon to select a process definition and specify parameter values.

Missed Actions – Specifies the action to perform if a scheduled task cannot run, as follows:

■ *Ignore* skips all missed scheduled tasks.

■ *System Alert* posts a system alert when a scheduled task is missed.

■ *Execute All* runs all missed scheduled tasks as soon as possible.

4. Click OK.

You have added the new task. It appears in the Scheduled Tasks list.

Edit a Scheduled Task

You can use the <u>Scheduler</u> (see page 41) to edit existing tasks that you want to run one or multiple times. Editing a scheduled task is useful to help implement procedures that have been updated or refined in your organization.

To edit a scheduled task

1. Click Administration, Tools, Scheduler.

   The Scheduled Tasks list appears.

2. Click the Edit (pencil) icon for the task that you want to update.

   The Edit Scheduled Task window appears, allowing you to modify the settings for the tasks.

3. Click OK

   The Catalog system saves your changes.

You have edited the scheduled task.

Delete a Scheduled Task

You can use the <u>Scheduler</u> (see page 41) to delete scheduling tasks that your organization no longer needs.

To delete a scheduled task

1. Click Administration, Tools, Scheduler.

   The Scheduled Tasks list appears.

2. Click the Delete (trash can) icon for each task that you want to delete.

   Each selected task is deleted and removed from the Scheduled Tasks list.

You have deleted the scheduled tasks.

Chapter 2: Managing Business Units and Accounts

This section contains the following topics:

Business Units

The organizational structure controls access to various data. The *business unit* is a branch in the organizational structure. A business unit can represent the following:

- A service provider (the highest level or *root* business unit)

- An external client company, if the service provider provides services to external customers

- An internal corporate department or group within a department

A business unit administrator can perform the following tasks:

- Access the catalog published by the business unit

- Subscribe accounts (see page 69) to services

- Perform other administrative duties for the business unit and its child business units

The following rules apply when you create your business unit structure:

- Business units can have child business units

- Users can have roles in more than one business unit

- Business units can have one or more accounts

- Each account can have many users

- Each user can have many accounts

- Each user who has an account can be billed to that account

- When you request a service, the parent business unit of your business unit defines the catalog that you see

- When you subscribe an account to a service, the parent business unit defines the catalog that you access (see page 46)

Default Business Units and Catalog Access

By default, users and accounts can request or subscribe to items *only* in the catalog of their parent business unit (see page 45).

Example:

Review the following example of a four-level hierarchy of business units, in a default configuration:

Service Provider (SP) is the top-level business unit.

A is a child business unit of SP.

B is a child business unit of A.

C is a child business unit B.

In linear form, you can express the relationship as follows: SP-->A-->B-->C. In this structure, the following apply:

■ Users and accounts in A can request or subscribe to items in the catalog of SP *only*, because A is a child of SP

■ Users and accounts in B can request or subscribe to items in the catalog of A *only*, because B is a child of A.

■ Users and accounts in C can request or subscribe to items in the catalog of B *only*, because C is a child of B.

To change the default behavior, use the following Service Builder configuration settings:

■ The System configuration settings named Use Service Provider Configuration Only and Use Service Provider Catalog Only

■ The Catalog configuration setting named Pass Through Catalog

Note: For instructions to modify configuration settings, see the *Implementation Guide*.

Tenant Administration

Deciding your organizational hierarchy is an important step in implementing CA Service Catalog. The business unit (also named *tenant*) organizational structure controls access to various data. The following requirements determine which methods of tenant administration you can use:

- If your organization has CA Service Desk Manager r12.5 installed, you can decide whether to create and maintain tenants using *common* tenant administration or *stand-alone* tenant administration.

- If CA Service Desk Manager is *not* installed, or if a CA Service Desk Manager release *earlier than* r12.5 is installed, then common tenant administration does *not* apply. Therefore, stand-alone tenant administration is the only method that applies.

If applicable, decide between *common* or *stand-alone* tenant administration, as follows:

- In *common* tenant administration, you create and maintain a single tenant structure for CA Service Catalog and CA Service Desk Manager. You create and maintain tenants, including parent and child tenants, in CA Service Desk Manager *only*.

  Thus, you can create, copy, cut, and paste tenants in the CA Service Desk Manager *only*. Similarly, you can edit common (shared) attributes (see page 69) of tenants in CA Service Desk Manager *only*. CA Service Catalog "inherits" the tenants, their structure, and their common attributes from the CA Service Desk Manager. These features appear as read-only in CA Service Catalog. In CA Service Catalog, you can still edit CA Service Catalog-specific attributes (see page 69).

- In *stand-alone* tenant administration, you create and maintain tenants, including parent and child tenants, in CA Service Catalog only, and these tenants apply only to CA Service Catalog. Tenant administration functions and tenant attributes are not shared between CA Service Catalog and any other product. If CA Service Desk Manager is installed, you create and maintain its tenant structures separately from CA Service Catalog. Stand-alone tenant administration is the default.

- Review the considerations for selecting a method of tenant administration (see page 48).

- Review the recommendations for selecting a method of tenant administration (see page 49).

- Review the recommendation concerning the movement tenants.

Considerations for Tenant Administration

Consider the following factors when you decide whether to use common or stand-alone tenant administration (see page 47):

- Common tenant administration helps ensure efficiency and consistency in your tenant structures for all applicable products providing a single point of administration. You can administer tenants once rather than multiple times in CA Service Catalog and CA Service Desk Manager.

- Common tenant administration saves costs for multiple service providers and other large organizations by providing easier, more efficient administration. You administer tenants at one place leveraging a common data model.

- Stand-alone administration provides flexibility for tenant structures across products. Thus, to use separate tenant structures for CA Service Catalog and CA Service Desk Manager, use stand-alone administration, not common tenant administration. Doing so is suitable in the following cases:

  - You have installed CA Service Catalog *without* CA Service Desk Manager.

  - You have installed CA Service Catalog with CA Service Desk Manager; *however*, you want to administer the business units for CA Service Catalog separately from these other products.

- If you have already implemented separate tenant structures in both products, you perform synchronization activities to enable common tenant administration.

Recommendation for Tenant Administration

We recommend the following when you decide whether to use common or stand-alone tenant administration (see page 47):

- Review the processes for configuring common tenant administration (see page 57) and for managing stand-alone business units (see page 49).

- Select and implement your choice (common tenant administration or stand-alone tenant administration) *before* creating tenants in CA Service Catalog and CA Service Desk Manager.

- Continue using your chosen option for the long term, preferably throughout the life cycles of the applicable products. While you can change options at any time, you *must* perform synchronization activities whenever you change from stand-alone tenant administration to common tenant administration.

- As a best practice for efficiency and consistency across applicable products, use common tenant administration.

Note: Common tenant administration has no effect on how you manage users and accounts.

Important! Whether you decide to use common or stand-alone tenant administration, we strongly recommend the following: Do *not* move a CA Service Catalog tenant (business unit), once any subscription or request exists for an associated account that belongs to the tenant. When such subscriptions exist, moving a tenant impacts its business rules and therefore may cause several problems in request management, user management, and other functions.

How to Manage Stand-Alone Business Units

You can create and maintain business units (tenants) using either common or stand-alone tenant administration (see page 47). Stand-alone administration means that you use CA Service Catalog alone to manage business units. When you use stand-alone administration, you can do the following in CA Service Catalog to manage business units:

- View a business unit (see page 50)

- Add a business unit (see page 50)

- Edit a business unit (see page 54)

- Manage the details of a business unit (see page 56)

- Delete a business unit (see page 56)

- Search for business units (see page 55)

View a Business Unit

Whether you manage stand-alone business units (see page 49) or use common multi-tenant administration (see page 57), you view business units and their structure. Viewing them helps you be familiar with them and helps you decide how to refine your business unit structure.

Follow these steps:

1. Select Administration, Business Units from the main menu.

   The Business Unit page appears. The page includes the business unit tree.

2. Expand the business unit tree to find the business unit that you want to view.

3. Click the Profile (down arrow) icon for that business unit.

   The Business Unit Profile appears.

4. View the fields as needed.

5. Do one of the following:

   ■ Click Back to return to the Business Unit page.

   ■ Click Edit to edit the business unit (see page 54).

You have viewed the business unit.

Add a Business Unit

When you manage stand-alone business units (see page 49), adding a tenant (business unit) is typically a required task as you create and refine your business unit structure. This function (adding a business unit) is disabled if you enable common multi-tenant administration (see page 57). When you enable it, you can *view* tenants and all their attributes in CA Service Catalog, but you can edit *only* the CA Service Catalog-specific attributes.

To add a business unit

1. Select Administration, Business Units from the main menu.

   The Business Unit page appears. The page includes the business unit tree.

2. Expand the business unit tree to find the parent business unit for the new business unit.

3. Click the (+) icon of the parent business unit.

   Note: The (+) icon appears *only* for business units that can have child business units.

   The Add a New Business Unit page appears.

4. Complete the fields for the new business unit on the Add a Business Unit page (see page 51).

5. Click OK.

   The Catalog system saves the new and updated data that you specified.

The business unit is created.

Add a Business Unit Page

When you add a business unit (see page 50), the Add a New Business Unit page appears. You complete the fields on this page to configure the new business unit to meet the needs of your organization.

The following fields on the Add a New Business Unit page require explanation:

Business Unit Login ID

Specifies the value you use to log in to a business unit (except the service provider business unit).

You specify this value in the Business Unit field of the CA Service Catalog login page to log in to a specific business unit. You also use this value when running ixutil import and export commands. This value must be unique.

Business Unit Name

Specifies the name of the business unit.

This field defaults to the same name as the Business Unit Login ID. The Business Unit Name appears throughout the user interface and in all reports and invoices.

Business Unit ID

Identifies an automatically generated, read-only value, whether you are using common or stand-alone tenant administration (see page 47).

After you create the business unit, CA Service Catalog automatically fills in this field. The only exception is the service provider business unit. You specify its business unit ID during the Service View installation and cannot change this ID afterwards.

If necessary, use the Business Unit ID field for reference. For example, you change your method of tenant administration from stand-alone to common (see page 47). In that case, you use the business unit ID when you create a common tenant mapping file (see page 59). You also use the business unit ID with ixutil and web services.

Opened Date

Specifies the date when the business unit becomes part of your organizational structure, in local time.

Currency Name, Date Format, Time Format, Decimal Symbol

Specify the currency name, date and time formats, and decimal symbol to appear on the user interface. These specifications appear, for example, when users create and manage requests. They also appear when administrators create and maintain services, users, and so forth.

Contains Sub Units

Specifies whether this business unit can have sub business units (child business units).

Note: You *cannot* change this setting after you have created the business unit.

Create Dashboard Library Namespace

Specifies whether to create a folder for the business unit in the Dashboard Library. Doing so helps organize the contents of the library.

Single Account Mode

Specifies whether the business unit can contain *only one* account.

If you select this option, specify the name of the single account.

Note: You *cannot* change this setting after you have created the business unit.

Create Document Namespace

Specifies whether to create a folder for the business unit in the Documents folder. Doing so helps organize the contents of the folder.

Primary Contact Information

Specifies the user ID of the primary contact for the business unit.

You can do the following:

- Click the Search (magnifying glass) icon to search for and select a user.

  When you select a user, the fields are filled with the available data for the selected user. Some fields can remain empty.

- Click the Remove (trash can) icon to clear the existing user data.

Theme

A *theme* specifies the settings for several look-and-feel elements, including images and icons (*except for* logos), menus, tabs, and so forth. When applicable, these elements include colors, font name and point size, highlighting, and related specifications. You customize these look-and-feel elements by editing the Cascading Style Sheet (CSS) files for the theme.

The look-and-feel of the UI matches the theme of the business unit that you are logged in to. If theme is not set for a business unit, CA Service Catalog checks the business unit hierarchy until it finds a theme. Thus, if a business unit does not have its own theme, it uses the theme of its closest parent business unit. You can use the same theme for all business units. Alternatively, you can optionally create and use different themes for different business units.

If you update a theme for a specific business unit, the change affects the users who belong to that business unit. The change also affects any child business units that do not have their own theme specified.

Note: For details about customizing themes, see the "Custom Branding" chapter in the *Implementation Guide*.

Logo

For each business unit, you can optionally specify a *business unit* logo. If you specify this logo, it replaces the *global* logo in the heading on product pages and request emails for users of the business unit. You can use a business unit logo to support the brand or other messaging uniquely for a business unit. You can update the logos for every business unit or only for specific business units. For example, you can decide to customize logos only for super tenants directly under the root business unit.

Important! If you have enabled multi-tenancy with CA Service Desk Manager, CA Service Catalog ignores any of its own settings for business logos. Instead, CA Service Catalog uses the logo or logos that the CA Service Desk Manager setup specifies, if applicable. If no CA Service Desk Manager logo applies, then each business unit uses the CA Service Catalog global logo.

If the business unit has child business units, the following applies:

- If the child business unit has its own logo specified, users who log in to it see the child logo, not the parent logo.

- If the child business unit does *not* have its own logo specified, users who log in to it see the global logo.

Thus, users with access to multiple business units can see different header logos when they log in to each business unit.

Note: For details about customizing logos, see the "Custom Branding" chapter in the *Implementation Guide*.

Location Information

Specifies details about the location of the business unit.

Note: All CA products using the same MDB share location data. Therefore, be careful when you edit location information.

You can do the following:

■   Click the Search (magnifying glass) icon to search for and select a location.

■   When you select a location, the fields are filled with the available data for the selected user. Some fields can remain empty.

■   Click the Remove (trash can) icon to clear the existing location data.

■   Click the Add (plus sign) icon to add a new location manually in the fields provided.

■   Click the Edit (pencil) icon to edit the location fields manually.

Edit a Business Unit

When you manage stand-alone business units (see page 49), editing a tenant (business unit) is typically a required task as you create and refine your business unit structure. This function (editing a business unit) is disabled if you enable common multi-tenant administration (see page 57). When you enable it, you can *view* tenants and all their attributes in CA Service Catalog, but you can edit *only* the CA Service Catalog-specific attributes.

To edit a business unit

1.   Select Administration, Business Units from the main menu.

     The Business Unit page appears. The page includes the business unit tree.

2.   Expand the business unit tree to find the business unit that you want to edit.

3. Click the Edit (pencil) icon for that business unit.

   The Edit Business Unit Profile page appears.

4. Edit the fields as needed. You can edit the same fields as when you complete the
   Add a Business Unit page (see page 51), *except as follows: Y*ou *cannot* update these
   fields.

   - Business Unit ID

   - Contains Sub Units

   - Single Account Mode

5. Click OK.

You have edited the business unit.

Search for Business Units

Whether you manage stand-alone business units (see page 49) or use common
multi-tenant administration (see page 57), you can search for business units. Doing so
helps you stay familiar with business units and helps you decide how to refine your
business unit structure. You can search for business units when you must perform an
action, such as invoice or update, on the accounts in that business unit.

To search for one more business units

1. Click Administration, Business Units.

   The Business Units page appears. This page includes the Business Units search filter.

2. Select the search mechanism, *one* of the following, and specify the related search
   criteria, as follows:

   - Standard Query

     Select the operator and specify the value.

   - Advanced Query

     Select the operator and specify the value for each criterion in the search.

   - Load Query

     Select the saved query to use.

   For Advanced Query and Load Query, the same guidelines apply as when you
   specify an advanced search for requests (see page 588).

3. Click Search.

4.   (Optional) Click Save Last Executed Query to save the last executed query.

5.   Click the Business Unit name in the Business Units list.

The business unit details page appears. You can search for accounts, maintain the business unit profile, and perform related tasks.

Delete a Business Unit

When you manage stand-alone business units (see page 49), deleting a tenant (business unit) is typically a required task as you maintain your business unit structure. This function (deleting a business unit) is disabled if you enable common multi-tenant administration (see page 57). When you enable it, you can *view* tenants and all their attributes in CA Service Catalog, but you can edit *only* the CA Service Catalog-specific attributes.

To delete a business unit

1.   Select Administration, Business Units from the main menu.

The Business Unit page appears. The page includes the business unit tree.

2.   Expand the business unit tree to find the business unit that you want to delete.

3.   Click the (-) Remove icon for the business unit.

A confirmation prompt appears.

4.   Confirm or cancel the deletion.

The business unit and its related data are deleted.

Manage the Details of a Business Unit

Managing the details of a tenant (business unit) is a required step in maintaining a tenant structure for CA Service Catalog. *Managing the details* refers to the tasks mentioned in the steps of this procedure. This function applies whether you use common or stand-alone tenant administration (see page 47). However, some parameters are read-only if you enable common multi-tenant administration (see page 57).

To manage the details of a business unit

1.   Select Administration, Business Units from the main menu.

The Business Unit page appears. The page includes the business unit tree.

2.   Expand the business unit tree to find the business unit you want.

3. Click the business unit name.

   The business unit management tabs appear.

4. Do one or more of the following:

   ■ Manage the accounts of the business unit

   ■ Edit the business unit profile

   ■ View summary information about the accounts in the business unit

You have managed the details of the business unit.

How to Configure Common Tenant Administration

Common tenant administration (see page 47) enables you to use a single administration tool to create and maintain business units (tenants) for multiple products, including CA Service Catalog, simultaneously. Common tenant administration is also named multi-tenant administration. To configure CA Service Catalog to use common tenant administration, follow this process:

Important! This topic applies *only* if your organization has CA Service Desk Manager installed; this topic does *not* apply to CA Service Desk Manager releases earlier than r12.5.

1. Meet the prerequisites (see page 58).

2. Configure the tenancy settings (see page 58) in CA Service Desk Manager to support the integration with CA Service Catalog.

   Important! CA Service Catalog requires a tenant of type Service Provider in CA Service Desk Manager, as a prerequisite for completing this process and its related procedures!

3. Create the common tenant mapping file (see page 59).

   In this file, you map tenants between CA Service Catalog and CA Service Desk Manager, so that both products use the same tenant structure.

4. Prepare to run the common tenant merge utility (see page 65).

   This utility uses the information in the common tenant mapping file to create a shared tenant structure between CA Service Catalog and CA Service Desk Manager.

5. Run the common tenant merge utility (see page 61).

6. Set the configuration variables for common tenant administration (see page 62).

7. As needed, manage the common tenants (see page 66), primarily using CA Service Desk Manager.

8. Optionally implement a common terms of use agreement (see page 67) for common tenants.

9. Note the effects of common multi-tenancy on business unit functions (see page 69).

Prerequisites

To enable common multi-tenant administration, meet the following prerequisites:

■ Verify that CA Service Desk Manager r12.5 is installed; the CA Service Desk Manager release *cannot* be earlier than r12.5

■ Verify that CA Service Catalog is installed

■ Verify that CA Service Catalog and CA Service Desk Manager share the same MDB; for details, see your CA MDB documentation

Configure the Tenancy Settings

When you configure common tenant administration (see page 57), a required task is configuring the tenancy settings in CA Service Desk Manager to support the integration with CA Service Catalog.

To configure the tenancy settings in CA Service Desk Manager to support the integration with CA Service Catalog

1. Log in to CA Service Desk Manager as ServiceDesk(administrator).

2. Click Administration, Option Manager, Multi Tenancy.

3. Verify that the multi-tenancy option is on.

4. Verify that the multi-tenancy depth is 10.

5. Click Administration, Security and Role Management, Tenants

6. Click Create New to create a tenant, if no tenant of type Service Provider exists. Enter a meaningful name for the new tenant.

7. Verify that Service Provider Checked is on and Subtenants Allowed is on.

   Important! CA Service Catalog requires a tenant of type Service Provider in CA Service Desk Manager, as a prerequisite for completing this process and its related procedures!

You have configured the tenancy settings in CA Service Desk Manager to support the integration with CA Service Catalog.

How to Create the Common Tenant Mapping File

When you configure common tenant administration (see page 57), a required task is creating the common tenant mapping file. To create the common tenant mapping file, follow this process:

1. Review the tenant structures in CA Service Catalog and CA Service Desk Manager, and plan the merged structure that you want to use for both. If necessary, map the structure by hand or by using modeling or graphics software. Keep the drawing or print version of the structure on hand for reference while completing these steps.

2. At the computer where the MDB is installed, using the Windows Control Panel, stop these CA Service Catalog services: CA Service View, CA Service Accounting, CA Service Fulfillment, and Message Queue 4.1 Broker.

3. Run the ant command to create the mapping file (see page 60).

4. Open the mapping file. After the Service Provider mapping entry, add a new line for each new mapping entry that you create. Specify each entry using the following format, using the Service Provider entry as a model:

   CA Service Desk Manager business unit ID=CA Service Catalog business unit ID

   For example:

   0x60B4EAC8B85E41DH97E647CF84A93CFA=87958EK983987D42AB2A4PAEF808C129

   Important! Mapping a parent tenant maps the parent *only*, *not* the children! Therefore, to cause the automatic mapping noted in the next step, do the following: *Either* map parents and children individually *or* omit the mapping of the parent.

5. Follow the rules for creating entries in the common tenant mapping file (see page 62).

6. Review the sample common tenant mapping file (see page 63) to become more familiar with the format and contents of the file and the results of running it. Use the sample file to help verify the contents of your mapping file.

7. Restart the CA Service Catalog services that you stopped at the beginning of this procedure.

You have created the common tenant mapping file. You are ready to run the utility.

Run the Ant Command to Create the File

When you configure common tenant administration (see page 57), a required task is creating the common tenant mapping file (see page 59). To create this file, you run the ant generate-merge-config command.

To run the ant command to create the file

1. Open the CA Service Catalog command prompt by clicking Start, Programs, CA, Service Delivery, Service Delivery Command Prompt.

2. At the CA Service Catalog command prompt, enter the following command:

   ant generate-merge-config

   The common tenant merge utility (see page 65) runs and creates the tenant mapping file, which is named merge-tenants.conf. The utility does the following in this file:

   - Creates the initial mapping entry, which maps the Service Provider business units for each product to each other; this entry is the only *required* entry. The CA Service Catalog entry is named ca_tenant, and the CA Service Desk Manager entry is named usm_tenant_ext.

   - Lists the names and database IDs of all CA Service Catalog tenants

   - Lists the names and database IDs of all CA Service Desk Manager tenants

   You use these names and database IDs to create the mapping entries.

3. Open the merge-tenants.conf file, using a text editor such as Notepad.

   This file is located in %USM_HOME%.

   If you open the file with a program that is not a text editor, save the file in text-only format.

4. Review the contents of the file and verify that the main sections listed earlier exist and appear appropriate for your enterprise. If they do not, verify that you have configured common multi-tenant administration (see page 57) *before* creating the mapping file. If necessary, verify your configuration and rerun the ant generate-merge-config command.

You have run the ant command to create the common tenant mapping file.

Run the Common Tenant Merge Utility

When you configure common tenant administration (see page 57), a required task is running the common tenant merge utility.

To run the common tenant merge utility

1. On the computer where the MDB is installed, open the CA Service Catalog command prompt by clicking Start, Programs, CA, Service Delivery, Service Delivery Command Prompt.

2. At the CA Service Catalog command prompt, enter the following command:

   ant merge-tenants

   The command runs. The utility prompts you for the information that you collected when you prepared to run this utility (see page 65).

3. Continue answering the prompts until the utility stops prompting you.

   If the utility detects one or more invalid mappings in the common tenant mapping file, the utility aborts, and you receive an error message. In such cases, correct the error in the file and run the utility again.

   Important! If the same name is used for two or more tenants being created in CA Service Desk Manager, a warning message is issued, and the utility aborts. In such cases, check each product's tenant structure, rename any duplicate tenant names to be unique, and re-run the utility.

   Otherwise, the utility runs and performs the following tasks:

   - Summarizes the changes to be made in each product's tenant table in the MDB before making the changes

   - Creates any missing tenants in those tables

4. Restart the CA Service Catalog services that you stopped at the beginning of this procedure.

5. Start CA Service Catalog and CA Service Desk Manager, and verify that the tenant structure and common attributes (see page 69) of tenants are the same in both products.

You have run the common tenant merge utility.

Set Configuration Variables

When you configure common tenant administration (see page 57), setting the related configuration variables is a required task.

To set configuration variables for common tenant administration

1. Log in to CA Service Catalog as a user with the Service Delivery Administrator role.

2. Click Administration, Configuration, System Information.

3. Verify that the value of the option named Common Tenant Data Synchronized is Yes.

   Note: This value is read-only. A value of No indicates that a discrepancy exists between the tenant structures of CA Service Desk Manager and CA Service Catalog. If this value is No, verify the accuracy of the common tenant mapping file and run the common tenant merge utility again.

4. Set the value of the option named Common Multi-Tenant Administration Enabled to Yes.

   When this setting is Yes, common multi-tenant administration is enabled for CA Service Catalog, through CA Service Desk Manager.

   Note: These settings are described in detail in the *Implementation Guide*.

You have set the configuration variables for common tenant administration.

Rules for Mapping File Entries

When you configure common tenant administration (see page 57), a required task is creating the common tenant mapping file (see page 59). Follow these rules when creating entries in this file:

Important! Violations of any of these rules cause the common tenant merge utility to abort.

■ Do not map business units that are at different levels. For example, do not map a first-level business unit in CA Service Desk Manager to a second-level business unit in CA Service Catalog. Similarly, do not map a seventh-level business unit in CA Service Desk Manager to a third-level business unit in CA Service Catalog.

■ Do not map the same business unit more than once. If you have several business units, print the merge-tenants.conf file and mark each business unit after you map it and save the file.

■ Do not map a child without mapping all its parents. Map the parents from the tenant level directly above the child through the level directly under the Service Provider business unit.

■ Verify that tenant names are unique across all products. For example, if you have a tenant named AAA in CA Service Desk Manager and CA Service Catalog, rename at least one of them to a unique, meaningful name.

■ Verify that CA Service Catalog has no duplicate tenant names within its own tenant structure. While CA Service Catalog can function with duplicate names, the other products cannot.

■ For every tenant and sub-tenant that you map, verify that the values in CA Service Desk Manager for common attributes meet your requirements for CA Service Catalog. Doing so is important because for common attributes, the CA Service Desk Manager values overwrite the CA Service Catalog values.

■ To add a complete tenant structure from one product to the other automatically, do *not* map that tenant or any of its sub-tenants explicitly in the merge-tenants.conf file.

Instead, run the common tenant merge utility and confirm that you want to add any tenants that are not mapped in the merge-tenants.conf file. The utility then adds the complete structures of any unmapped tenants to each product automatically.

Sample Common Tenant Mapping File

When you configure common tenant administration (see page 57), a required task is creating the common tenant mapping file (see page 59). Review the following sample common tenant mapping file to help you construct your own file:

```
…
# CA Service Catalog Tenants (id, name)
# ca.222.com    ca.222.com
# 00234A51DC4A4F70A03D3BDE5526278C            BB
# 9F6309A0CB654781B08080AD78C2280F    CC
# 9CF5655B4B8D4833A0C6E74EB56128C5            AA
#
# CA Service Desk Manager Tenants (id, name)
# 0xB3484A535A3D994B9FBCD28D3845F292    ca.111.com
# 0xBB83719AA93DFC48B909D7D72CF8B0CB    A
# 0xD9DC8FF2EB84CC43988FE71F4B489D3D    E
# 0x67958EA983987D42AB2A4BAEF808C029    D
#
# Service Provider tenants must map to each other. This mapping is mandatory.
0xB3484A535A3D994B9FBCD28D3845F292=ca.222.com
#
# Map A to AA
0xBB83719AA93DFC48B909D7D72CF8B0CB=9CF5655B4B8D4833A0C6E74EB56128C5
…
```

In this example, the following tenants exist:

■ CA Service Desk Manager tenants: ca.111.com, A, D, and E

■ CA Service Catalog tenants: ca.222.com, AA, BB, and CC

Service Provider tenants must map to each other. For this reason, you merge the CA Service Desk Manager Service Provider tenant (ca.222.com) and the CA Service Catalog Service Provider tenant (ca.111.com).

Otherwise, you add a mapping entry only when you want to merge two tenants together. For example, in the sample file, you merge CA Service Desk Manager tenant A and CA Service Catalog tenant AA.

When you merge tenants, the CA Service Desk Manager tenant overwrites all common attributes in the CA Service Catalog tenant, but CA Service Catalog-specific attributes remain intact. Thus, after you run the common tenant merge utility with this sample file, the tenant ca.222.com in CA Service Catalog is renamed ca.111.com. Similarly, tenant AA in CA Service Catalog is renamed A. Each of these tenants also inherits the other common attributes (see page 69) of the CA Service Desk Manager tenant with which it was merged.

If you are simply adding tenants from one product to the other, you do not need to map them in the common tenant mapping file. For this reason, the sample file does not map any other tenants. Thus, the following changes occur when you run the common tenant merge utility with this sample file:

■ CA Service Catalog tenants BB and CC are added to CA Service Desk Manager

■ CA Service Desk Manager tenants E and D are added to CA Service Catalog

Prepare to Run the Common Tenant Merge Utility

When you configure common tenant administration (see page 57), a required task is preparing to run the common tenant merge utility.

To prepare to run the common tenant merge utility

1. Collect the following information and have it ready when you run the utility:

    ■ The name of the common tenant mapping; the default name is merge-tenant.conf

    ■ The location of the common tenant mapping file; the default location is %USM_HOME%

    ■ The password for any administrator with the Service Provider role in CA Service Catalog. An example is the default user named spadmin.

      If running the utility creates tenants in CA Service Desk Manager, you enter this password.

2. Shut down CA Service Desk Manager and CA Service Catalog.

3. Stop these CA Service Catalog services on the computer where the MDB is installed: CA Service View, CA Service Accounting, CA Service Fulfillment, and Message Queue 4.1 Broker. Using the Windows Control Panel to stop the services.

4. Verify that the common tenant mapping file (see page 59) is complete, and verify its location.

You have prepared to run the common tenant merge utility.

How to Manage Common Tenants

When common tenant administration (see page 57), is enabled, all tenants are *common* tenants; they exist in CA Service Catalog and also exist in CA Service Desk Manager. You can manage these tenants using CA Service Desk Manager, but not CA Service Catalog. Here, *managing* means adding or deleting tenants, or editing their common attributes. In CA Service Catalog, you can *view* tenants and all their attributes, but you can edit only the CA Service Catalog-specific attributes. To manage common tenants, follow this process:

1. In CA Service Desk Manager, do the following:

   ■ Create one or more new tenants, as needed.

   ■ Make inactive any tenants that you no longer want to use, if applicable.

   ■ Edit the common attributes, such as name, description, contact, location, inactive, and so forth.

   ■ Verify the CA Service Desk Manager and CA Service Catalog tenant and business unit settings (see page 66). Update these settings, if necessary. These settings enable the tenant mapping between the products.

   Note: To begin some of these tasks, select Administration, Security and Role Management, Tenants, Start. For instructions to perform these tasks, see the CA Service Desk Manager documentation.

2. In CA Service Catalog, view the business units by clicking Administration, Business Unit, and do the following:

   ■ Verify that any new business units (tenants) that you created in CA Service Desk Manager are visible in CA Service Catalog.

   ■ Verify that any business units that you made inactive in CA Service Desk Manager are no longer visible in CA Service Catalog.

   ■ Verify that you cannot add or delete business units or edit their primary attributes in CA Service Catalog.

3. Note the effects of common multi-tenancy on business unit functions (see page 69).

You have managed the common tenants.

Tenant and Business Unit Settings

Verify the CA Service Desk Manager tenant and CA Service Catalog business unit settings; these settings enable the tenant mapping between the products.

Tenant Setting for CA Service Desk Manager	CA Service Catalog Business Unit Setting
Service Provider selected (on)	Service Provider (SP), meaning root or highest level business unit

Subtenants Allowed selected (on)	Contains Sub Units: True, meaning Super Tenant. A super tenant is a tenant with at least one subtenant.
Subtenants Allowed not selected (off)	Contains Sub Units: False, meaning Tenant, also named leaf (end-of-tree) tenant. A subtenant is a tenant with at least one parent tenant.
Inactive Tenant	inactive, formerly deleted tenant
Parent Tenant is Empty	SP is the parent tenant

How to Implement Common Terms of Use Agreement

As part of configuring CA Service Catalog to use common tenant administration (see page 57), you can optionally implement a common terms of use agreement for CA Service Catalog and CA Service Desk Manager. Doing so helps enforce a consistent login policy for all common tenants in both products. CA Service Desk Manager administrators create and maintain this agreement in CA Service Desk Manager. CA Service Catalog administrators configure CA Service Catalog to adopt this agreement. The effect on users attempting to log in to CA Service Catalog depends on the terms of use agreement settings in CA Service Desk Manager.

To configure CA Service Catalog to implement the terms of use agreement settings in CA Service Desk Manager, follow this process.

Note: For details about configuring terms of use agreement in CA Service Desk Manager, see the CA Service Desk Manager documentation.

1. In CA Service Catalog, select Administration, Configuration, System Information.

2. Specify Yes for the setting named Terms Of Usage Prompt Enabled.

    Note: If you specify No, then users attempting to log in to CA Service Catalog are *not* prompted to accept the terms of use agreement, regardless of the CA Service Desk Manager setting.

3. Review the effect of the CA Service Desk Manager setting on users logging in to CA Service Catalog.

   The effect depends on the terms of use agreement configured for the common tenant in CA Service Desk Manager, as follows:

   ■ If the common tenant has an active, defined (not empty) terms of use agreement, prompt the user to accept it at every login.

   ■ If the common tenant has no terms of use agreement defined, or if the agreement is inactive, do the following: Review the parent hierarchy of the tenant, and do the following:

     – If the nearest parent has active but empty terms of use agreement defined, then do not prompt the user with any terms of use agreement. Otherwise, prompt the user to accept the terms of use agreement of the nearest parent tenant that has an active terms of use agreement defined.

     – If no parent tenant has an active, specified terms of use, do not prompt the user with any terms of use.

   ■ If the common tenant has an active terms of use agreement defined but is configured to suppress it, then do *not* prompt the user to accept it.

   If users attempting to log in to CA Service Catalog receive the prompt to accept the terms of use agreement but do not accept it, they cannot access CA Service Catalog.

4. Verify that the CA Service Desk Manager setting meets the needs of your organization. If necessary, see the CA Service Desk Manager administrator or documentation for assistance to update the CA Service Desk Manager setting.

Effects of Common Tenant Administration

After you have completed configuring common tenant administration (see page 57), the following results occur in CA Service Catalog:

- A new configuration item appears on the Administration, Configuration page. The name of this item is Common Multi-Tenant Administration. Its default setting is No.

  When you set this option to Yes, the following changes occur:

  – The *common attributes* of the tenant between CA Service Catalog and CA Service Desk Manager, become read-only from CA Service Catalog. You can edit these common fields in CA Service Desk Manager *only*:

    - The Business Unit Name and Description in the General Information section

    - All fields in the Primary Contact Information section

    - All fields in the Location Information section

  – You can edit the *CA Service Catalog-specific attributes* in CA Service Catalog. These attributes include the following fields:

    - Federal Tax Payer ID, State Tax Payer ID, Tax Region

    - Timezone, Currency Name, Date Format, Time Format

    - Decimal Symbol, Opened Date, Email, Web Site

  – The options to add, delete, edit, cut, and paste tenants are disabled.

  – A message appears on the Administration, Business Units page explaining why these functions are disabled.

- If you implement a common terms of use agreement (see page 67), then the terms of use settings in CA Service Desk Manager govern the attempts of users to log in to CA Service Catalog.

Accounts

Accounts are important components of a business unit (see page 45). An account can represent a corporate unit, a regional office, an individual user, or any logical grouping of them. CA Service Accounting applies charges to accounts.

The Catalog system uses accounts to request services and to subscribe to services. If a user (see page 77) requests a service from the catalog, the Catalog system automatically does the following:

- Creates an account for the user

- Subscribes the account to the requested service

How to Manage Accounts

You manage the accounts for a business unit by performing the following tasks

- Adding accounts (see page 70)

- Editing accounts (see page 72)

- Closing accounts (see page 73)

- Deleting accounts (see page 73)

- Searching for accounts (see page 74)

- Managing the requests for an account, using the catalog functions for handling requests pending action (see page 604)

- Managing subscriptions (see page 409), using CA Service Accounting

- Managing invoices (see page 419), using CA Service Accounting

- Viewing accounting profiles (see page 407) and modifying accounting profiles (see page 408), using CA Service Accounting

More information:

Using CA Service Accounting (see page 387)
Using CA Service Catalog for Catalog Users (see page 561)

Add an Account

An account (see page 69) can represent a corporate unit, a regional office, an individual user, or a logical group of them. You can add an account so that you can bill it for services that it has requested or subscribed to.

To add an account

1. Select Administration, Business Units from the main menu.

2. Expand the business unit tree to find the business unit to which you want to add an account.

3. Click the business unit name

4. Click Add Account.

   The Add a New Account page appears.

5. Complete the fields on the Add a New Account page (see page 71).

6. Click OK.

The account is created.

Add a New Account Page

When you add an account (see page 70), the Add a New Account page appears. You complete the fields on this page to configure the new account to meet the needs of your organization.

The following fields on the Add a New Account page may require explanation:

Opened Date

Specifies the date when the account becomes part of your organizational structure, in local time.

Status

Specifies whether the account is open or closed.

Note: A closed account cannot request or subscribe to a service. You cannot run invoices for a closed account after the final billing run is completed for the billing period during which the account was closed.

Primary Contact Information

Specifies the user ID of the primary contact for the account.

You can do the following:

■ Click the Search (magnifying glass) icon to search for and select a user.

■ When you select a user, the fields are filled with the available data for the selected user. Some fields may remain empty.

■ Click the Remove (trash can) icon to clear the existing user data.

Location Information

Specifies details about the location of the account.

Note: All CA products using the same MDB share location data. Therefore, be careful when you edit location information.

You can do the following:

■ Click the Search (magnifying glass) icon to search for and select a location.

■ When you select a location, the fields are filled with the available data for the selected user. Some fields may remain empty.

■ Click the Remove (trash can) icon to clear the existing location data.

■ Click the Add (plus sign) icon to add a new location manually in the fields provided.

■ Click the Edit (pencil) icon to edit the location fields manually.

Account Settings

Specifies the list of users associated with this account. Usage-based billing with CA Service Accounting references these users.

You can do the following:

■   Click the Search (magnifying glass) icon to search for and add a user.

■   Select an existing user in the list and click the Remove (trash can) icon. This action removes the user from the account.

■   Click the Add (plus sign) icon to add a new location manually in the fields provided.

Note: When a user creates a request for the first time, the Catalog system creates a request-related account for the user.

Edit an Account

Editing an account (see page 69) is typically a required task as you create and refine accounts in your business unit over time.

To edit an account

1.   Select Administration, Business Units from the main menu.

2.   Expand the business unit tree to find the business unit that contains the account you want to edit.

3.   Click the business unit name to display the list of accounts for the business unit.

4.   Click the account name that you want to edit.

The Account Profile page appears.

5.   Click Edit.

The Edit Account Profile page appears.

6.   Edit the same fields as on the Add a New Account page (see page 71).

7.   Click OK.

The account is updated with your edits.

Close an Account

Closing an <u>account</u> (see page 69) is typically a required task as you create and maintain accounts in your business unit over time.

To close an account

1. Review the <u>effects of deleting an account</u> (see page 74); the same effects apply when you close the account. Verify that you want to close the account.

2. Run the final <u>invoicing</u> (see page 394) for the account, if necessary.

3. Select Administration, Business Units from the main menu.

4. Expand the business unit tree to find the business unit that contains the account you want to close.

5. Click the account name that you want to edit.

6. Edit the account profile and change the status to Closed.

7. Click OK.

You have closed the account.

Delete an Account

Deleting an <u>account</u> (see page 69) is typically a required task as you maintain accounts in your business unit over time.

To delete an account

1. Review the <u>effects of deleting an account</u> (see page 74) and verify that you want to delete it.

2. Verify whether you must perform final billing for the account. If Yes, do the following:

    a. <u>Close the account</u> (see page 73).

    b. Run the final <u>invoicing</u> (see page 394) for the account.

3. Select Administration, Business Units from the main menu.

    The Business Units page appears.

4. Expand the business unit tree to find the business unit that contains the account you want to delete.

5. View the account profile and click Delete.

You have deleted the account.

Effects of Deleting an Account

Deleting an account (see page 73) affects the subscriptions and requests for the account, as follows:

Subscriptions

The status of subscriptions for the account change to the default cancellation status (Pending Cancellation or Cancel).

Requests

The status of requested service options for the account change based on their original status as follows:

Original Status	New Status
Not Submitted	None; the request is deleted
Submitted, an approval status, a fulfillment status, Pending Resource Assignment, or Resource Assigned	Cancelled
Completed	The default cancellation status (Pending Cancellation or Cancel) if CA Service Accounting is installed; Cancelled if CA Service Accounting is *not* installed
Pending Cancellation or Cancelled	same as original status

Search for Accounts

You can search for accounts, especially when you must perform an action, such as invoice or update, on the accounts.

To search for accounts

1. Click Home, Search.

   The Search page for accounts, invoices, and payments appears.

2. Click Accounts (the default) under the Search menu, if it is not selected already.

   The accounts search filter appears.

3. Select the search mechanism, *one* of the following, and specify the related search criteria, as follows:

   ■ Standard Query

      Select the operator and specify the value.

   ■ Advanced Query

      Select the operator and specify the value for each criterion in the search.

   ■ Load Query

      Select the saved query to use.

   For Advanced Query and Load Query, the same guidelines apply as when you specify an .

4. Click Search.

5. (Optional) Click Save Last Executed Query to save the last executed query.

6. Click the Account name in the Accounts list.

   The account profile appears.

Chapter 3: Managing Users and Roles

This section contains the following topics:

Users

A user in CA Service Catalog typically represents a person who uses the product. A person must have a user ID to log in and use CA Service Catalog. Each user requires a role (see page 89) based on the business unit specified during login. If the user does not specify a business unit at login, CA Service Catalog assigns the default role for all users (see page 94).

Other CA products sharing the same MDB as CA Service Catalog share some data with CA Service Catalog. For example, CA Service Catalog and other CA products share users and contacts. Some of these users apply to other CA products only (see page 77) (not CA Service Catalog). Therefore, use caution when managing users, to help protect data shared among multiple CA products using the same MDB.

Also, some data in the MDB is product-specific and is not shared between products. For example, CA Service Catalog does not share role information.

User Data in the MDB

CA Service Catalog and other CA products that use the same MDB share some of the same data, including users (see page 77).

The MDB automatically includes several users that CA Service Catalog does *not* use but that *other CA products* do use. These users include the following: System_ADH_generated, System_AM_User, Systemt_Anonymous, System_Argis_User, System_MA_User, System_NSM_generated, System_SD_User and UAPM Administrator.

User Data in CA EEM

The user ID of each CA Service Catalog user (see page 77) maps to a matching user in CA EEM. CA EEM authenticates and validates the CA Service Catalog user ID. You can optionally configure CA EEM to use an external directory, for example, Microsoft Active Directory. In this case, CA Service Catalog uses the user ID from Active Directory, and Active Directory validates the password.

The MDB stores user IDs and all other user data *except* passwords. When a user attempts to log in to CA Service Catalog, it passes the user ID to CA EEM for authentication. If CA EEM does *not* use an external directory, CA EEM authenticates the user ID. If CA EEM *does* use an external directory, the external directory authenticates the user ID.

If CA EEM uses an external directory, when an administrator creates a user, the Catalog system fetches important user details from the external directory. Examples include the first name, last name, email address, and organizational hierarchy. This fetching helps create the user quickly and efficiently.

CA EEM also controls access for each role (see page 89) to CA Service Catalog components.

Synchronization of User Data in CA EEM

If you use an external directory to maintain CA Service Catalog users (see page 77), do the following:

- Configure CA EEM to use an external directory
- Regularly synchronize the MDB users with the external directory users

Performing these tasks helps ensure that all CA EEM users have matching CA Service Catalog users.

Note: For details on synchronizing MDB users with external directory users and on configuring CA EEM and NTLM authentication, see the *Integration Guide*.

Single Sign-On

Single sign-on is one option for authenticating CA Service Catalog users (see page 77). If your organization uses a Windows domain, you can optionally configure CA Service Catalog to use single sign-on through Windows NTLM authentication.

Note: For details about using single sign-on through Windows NTLM authentication, see the *Implementation Guide*.

How to Manage Users

You can manage users (see page 77) within the scope of your role, as follows:

■ Search for users (see page 79)

■ View a user profile (see page 80)

■ Review the requirements and considerations when you add users (see page 80)

■ Add users (see page 81)

■ Set the format of automatically created passwords (see page 85)

■ Edit users (see page 86)

■ Requirements and considerations when you edit users (see page 86)

■ Delete users (see page 87)

Search for Users

You can search for users (see page 77), especially when you must perform an action, such as update the role or other data for a user.

To search for users

1. Click Administration, Users.

    The Users page appears. The User ID search filter appears at the top of the page.

    Note: By default, no users are shown in the search results list.

2. Select the search mechanism, *one* of the following, and specify the related search criteria, as follows:

    ■ Standard Query

       Select the operator and specify the value.

    ■ Advanced Query

       Select the operator and specify the value for each criterion in the search.

    ■ Load Query

       Select the saved query to use.

    For Advanced Query and Load Query, the same guidelines apply as when you specify an advanced search for requests (see page 588).

    Note: The user list can be large, especially if CA EEM has been configured to use an external directory. Therefore, use selective search criteria.

3. Click Search.

    The search results appear.

4. (Optional) Click Save Last Executed Query to save the last executed query.

5. Click the user name in the user list.

   The user profile appears.

You can view and optionally edit (see page 86) or delete (see page 87) the user.

View a User Profile

You can view profiles for users (see page 77), especially when you want to verify information before updating a user.

To view a user profile

1. Search for the user (see page 79) whose profile you want to view.

   The search results appear.

2. Click the user name or Profile icon of the user you want to view.

   The User Profile window appears.

You have viewed the use profile.

Requirements and Considerations When You Add Users

When you add a new user (see page 77) to CA Service Catalog, also define the user to CA EEM, so that the user can log in to CA Service Catalog. Use one of the following methods to do so:

■ If CA EEM uses an external directory such as Microsoft Active Directory, then its administrator must define the user in the external directory. For instructions, see the documentation for the external directory.

■ If CA EEM uses its own repository, then CA EEM automatically creates a new Global user in the USM/users folder when you add a new CA Service Catalog user. The Create EEM User action on the User Create event rule sets this behavior. You can optionally do one of the following:

   – Disable the action (see page 36) that creates the CA EEM user automatically

   – Set the format of automatically created passwords (see page 85) in CA EEM

Add a User

You typically add a [user](#) (see page 77) when a new employee starts working in your organization.

To add a user

1. Select Administration, Users from the main menu.

2. Click the Add button to add a new user.

   The Add a New User page appears.

3. Enter the data for the new user on the [Add a New User page](#) (see page 81).

4. Click OK.

The user is added.

More information:

[Roles and Default Access Rights](#) (see page 89)

Add a New User Page

You complete several fields when you [add a new user](#) (see page 81) in your organization. The following fields require explanation:

User ID

Specifies the ID by which CA Service Catalog identifies the user.

The User ID value must be unique.

If CA EEM is configured to use an external directory, data is auto-populated from the external directory data, if applicable.

Manager

Specifies the manager for the user being creating. To specify a manager, click the Search (magnifying glass) icon to display the list of users and select one. You can optionally click the Search icon in this list to specify advanced search criteria. Clicking that icon switches between advanced and standard search criteria.

To clear the Manager field, click the minus icon.

When your organization uses system approval as the [approval process](#) (see page 510), the manager must approve requests submitted by this user. If your organization uses another approval process, the manager can be a required approver, depending on how you configure the approval process.

Request Auto-Delegation: Delegate

Specifies the user to whom your requests pending action are delegated automatically when you auto-delegate your own requests pending action (see page 621). In addition, administrators can auto-delegate the requests pending action of other users (see page 623).

To specify a delegate, click the Search (magnifying glass) icon to display the list of users and select one. You can optionally click the Search icon in this list to specify advanced search criteria. Clicking that icon switches between advanced and standard search criteria.

To clear the Request Auto-Delegation: Delegate field, click the minus icon.

When you clear this field in your own user profile, your requests pending action stop being auto-delegated and remain in your queue. When you clear this field in the profile of another user, the requests pending action of that user stop being auto-delegated. The requests remain in the queue of that user.

Clearing this field *does not affect* requests pending actions that were already delegated to previously assigned delegates. Therefore, as a best practice, after clearing this field, instruct the former delegates to handle requests pending action (see page 604) (for example, approve, reject, or transfer) promptly. Alternatively, as an administrator, you can also handle them yourself or transfer them to other users (see page 618).

Delegate Use of Catalog: Delegates

Specifies the users to whom you delegate the use of your catalog (see page 629). These users can create and submit request from your catalog on your behalf. In addition, administrators can delegate the catalog of one user to another user.

This field is valid only if you or another administrator has enabled delegation of catalogs (see page 634) for your business unit.

To specify a delegate, click the Search (magnifying glass) icon to display the list of users. You can optionally click the Search icon in this list to specify advanced search criteria. Clicking that icon switches between advanced and standard search criteria. Select one or more users, and click the Plus sign (+) to save your selections.

To clear the Delegate Use of Catalog: Delegates field, click the minus icon.

When you clear this field in your own user profile, your catalog is no longer delegated. Also, your former delegates can no longer create and submit requests on your behalf from your catalog. When you clear this field in the profile of another user, the catalog of that user is no longer delegated. Also, the former delegates of that user can no longer create and submit requests on behalf of that user from the catalog of that user.

User Location

Specifies details about the location for the user.

Note: All CA products using the same MDB share the same location; therefore, use caution when modifying the location.

You can do the following:

■   Click the Search (magnifying glass) icon to list the existing locations and select one. You can optionally click the Search icon in this list to specify advanced search criteria. Clicking that icon switches between advanced and standard search criteria.

■   Clear the existing location by clicking the Remove (trash can) icon.

■   Add a new location by clicking the New Location icon and entering details.

■   Edit the assigned location by clicking the Edit (pencil) icon and entering details.

Select Business Unit

Specifies the business unit (see page 45) of the new user.

The default is the current business unit (the one that you are currently logged in to).

You require the Service Delivery Administrator role or Super Business Unit administrator role to change the business unit of the new user. Otherwise, you *can* create the user but *cannot* change its business unit.

In addition, users can belong to multiple business units. However, a user can have only one role and one authorization level in each business unit.

To select the business unit for either the role or authorization level, click the Search (magnifying glass) icon to display the list and select one. The icon appears to the left of the Select Business Unit column for either the role or authorization level. Click the name of the business unit to select it. You can optionally click the Search icon in the list of business units to specify advanced search criteria. Clicking that icon switches between advanced and standard search criteria.

Select Role

Specifies the role (see page 89) for the new user in the current business unit.

By default, new users receive the default role for all users (see page 94). However, administrators can optionally assign a different role. When you select a role, keep in mind the relationship between users, roles, and login (see page 95).

Select an available role and click the *Add Row* icon to add the role to the list of roles for the user.

(Optional) Select Authorization Level

Specifies the authorization level (see page 84) of the new user. This setting applies *only* if you use system approval as the approval process (see page 510).

Authorization Level

When you add a user (see page 81) or update a user, you can specify an authorization level for each business unit in which the user has a role. The system approval process is the only approval process (see page 510) that uses the authorization level. The authorization level does not apply to any other approval process. By default, the following levels are available with the text and numeric values shown:

- Level 0 (0)

- Level 10 (10)

- Level 20 (20)

- Level 30 (30)

- Level 40 (40)

- Level 50 (50)

In the system approval process, each *user* has an authorization level and each *service* has an approval level. The Catalog system automatically approves requests from users whose authorization level matches or exceeds the approval level of the service. Otherwise, the system follows a managerial hierarchy. The system routes the request to one or more request managers until it reaches one whose authorization level matches or exceeds the approval level of the service.

For example, if a service has an approval level of 40, then a user requesting the service requires one of the following:

- The user must have an authorization level of 40.

- The user must obtain approval from a request manager who has an authorization level of 40.

Note: The Requested For user can have different roles–and therefore different authorization levels–multiple business units. Therefore, the Catalog system uses the authorization level of that user in the business unit to which the requested *service* belongs.

Set the Format of Automatically Created Passwords

You can configure CA EEM to use its own repository (not an external directory). In that case, CA EEM automatically creates a user with a password when you either add a new CA Service Catalog user or update its user ID field. By default, the password is the same as the user ID. To meet the standards or conventions of your company, department, or customer, you can optionally change the format of the passwords created automatically in CA EEM.

To set the format of the passwords created automatically in CA EEM

1. Select Administration, Tools, Events.

2. For the User Create event, disable the Create EEM User rule and create a rule to use in its place.

3. Disable the associated rule action named Create New EEM User or Modify EEM User, and create an action to use in its place.

4. Open your replacement action and locate the Java class field. This field appears similar to the following:

    com.ca.usm.ruleEngine.action.CreateEiamUserAction,passwordTemplate=*value*

    The passwordTemplate parameter specifies the text to be used as the initial password. If the passwordTemplate parameter is empty, then the password is set to the user ID being created.

5. Set this parameter to a specific text value or to an available event variable.

    To insert an event variable in the Java Class field, click the arrow icon and select the variable from the list displayed. For example, to set the password to the user ID followed by the letter *a*, specify the following:

    com.ca.usm.ruleEngine.action.CreateEiamUserAction,passwordTemplate=$user_id$a

You have configured the automatically generated passwords to use the specified format.

Requirements and Considerations When You Edit Users

Follow these requirements and considerations when editing users:

- If a user ID exists in CA Service Catalog, verify that the user ID also exists in CA EEM. If not, create the user ID in CA EEM.

- If CA EEM uses an external directory such as Microsoft Active Directory, then its administrator must edit the user in the external directory. For instructions, see the documentation for the external directory.

- Running the synchronization utility (see page 78) synchronizes all user data, such as first name, last name, user name, password, from the external directory to the MDB. Running the utility also activates and deactivates users, when applicable.

- If CA EEM uses its own repository, when the User ID field for a CA Service Catalog user is newly populated (from being previously empty), the following occurs: A new CA EEM Global user is automatically created in the USM/users folder. If you update the User ID field, then the corresponding CA EEM user ID value is also updated with the same change.

  The Modify EEM User action on the User Change event rule governs this process. The rule action creates the CA EEM User ID with an initial password set to be the same as the user ID. Therefore, users must change their passwords when they log in for the first time. If the old user exists in CA EEM but its user ID has changed, the user ID in CA EEM is updated. You can optionally do one of the following:

  - Prevent the CA EEM user from being updated automatically by disabling the associated rule or action.

  - Set the format of automatically updated passwords (see page 85) in CA EEM.

Edit a User

You typically edit a user (see page 77) for any of several possible reasons. Examples include an existing employee changing roles or titles or transferring to a new department.

To edit a user

1. Select Administration, Users from the main menu.

   The Users page appears.

2. Search for the user (see page 79) you want to edit.

   The search results appear.

3. Click the Edit icon for the user you want to edit.

   The Edit User page appears.

   You can only edit users who have a role in the business units within the scope of your role.

   Important! Do *not* change the user ID of a user. Do *not* reuse user IDs that have been deleted, because all deleted users are maintained as *inactive* in the user database.

4. Update values as needed. You can edit the same fields as on the Add a New User page (see page 81).

   The fields are updated.

5. Click OK when you have completed all required fields.

The user is updated.

More information:

Add a User (see page 81)

Delete a User

When users (see page 77) leave your organization or no longer need access to CA Service Catalog, remove their user names from CA Service Catalog. Doing so helps maintain the product.

To delete a user

1. Review the effects of deleting a user (see page 88) and verify that you want to delete the user.

2. Select Administration, Users from the main menu.

   The Users page appears.

3. Search for the user (see page 79) you want to edit.

   The search results appear.

4. Click the Delete icon of the user you want to delete. You can delete only users who have a role in the business units within the scope of your role.

   The CA Service Catalog user is deleted. The related CA EEM user is *not* deleted.

   Note: You can delete one or more users by selecting the check boxes next to their user names and then clicking the Delete button.

You have deleted the user.

Effects of Deleting a User

Deleting a user (see page 87) affects its account, including its subscriptions and requests, as follows:

Account

Because future transactions can exist for an account, the account remains open when you delete the user. As an administrator, you can optionally close the account or leave it open.

Subscriptions

The status of subscriptions for the deleted user change to the default cancellation status (Pending Cancellation or Cancel).

Requests

The status of requested service options for the deleted user change based on their original status as follows:

Original Status	New Status
Not Submitted	None; the request is deleted
Submitted, an approval status, a fulfillment status, Pending Resource Assignment, or Resource Assigned	Cancelled
Completed	The default cancellation status (Pending Cancellation or Cancel) if CA Service Accounting is installed; Cancelled if CA Service Accounting is *not* installed
Pending Cancellation or Cancelled	same as original status

User Groups

You or other administrators can organize CA Service Catalog users (see page 77) into the following types of user groups:

- Global and Application user groups in CA EEM

  Two examples of application groups that CA Service Catalog uses are the CA WF Admin User or Super User.

  Note: For information about creating these groups and assigning users to them, see your CA EEM documentation.

- User-defined groups in CA EEM

  You create user-defined groups, in either your external directory (if applicable) or CA EEM. You can apply the same action to all users in the user-defined group, instead of modifying the users individually. Once you create a user-defined group in CA EEM or your external directory, it is available in CA Service Catalog.

  Note: For information about creating user-defined groups and assigning users to them, see the *Integration Guide*.

User group memberships appear as *read-only* data in user profiles (see page 80).

Roles and Default Access Rights

Each user (see page 77) can have a different role for each business unit. By default, the available roles provide access to different types of functionality in CA Service Catalog. However, the following factors greatly influence the functions that users can perform, regardless of their role:

- This topic describes the default access rights predefined by the product for each role. However, keep in mind the following:

  – Service Delivery administrators can change some default access rights for the *entire* Catalog system as follows: Log in to the root (highest level) business unit, select Administration, Configuration, and change the Access Control configuration settings.

  – Service Delivery administrators and business unit administrators can change several of these default access rights for *specific business units* as follows: Log in to the business unit, select Service Builder, Configuration, and change the Access Control configuration settings.

  Note: For details about the Access Control configuration settings, see the *Implementation Guide*.

■ All users can also <u>delegate the use of their catalogs</u> (see page 629) to other users to create requests on their behalf.

■ The Catalog system creates only one user at installation time. This user, named spadmin, has the Service Delivery Administrator role.

■ The table of <u>tasks that each role can perform</u> (see page 92) lists the tasks that each role can and cannot perform, based on the default settings.

Request-related functionality is available when CA Service Catalog is installed. Subscription and invoice-related functionality is available when CA Service Accounting is installed

Catalog User

Is the user role for requesting services *without* subscriptions. These users can also manage their own requests, such as approve, reject, fulfill, and other actions to <u>handle requests pending action</u> (see page 604).

Most users in the organization use this role *only*.

This role is predefined as the default role for new users. However, administrators can optionally change the default role for new users from the catalog user to another role.

This role is most suitable when you are *not* using subscriptions or billing in your implementation.

End User

Is end user for all functions available through the catalog. This user includes all the same access rights as the catalog user. In addition, the end user can subscribe to services, view invoices, view and add news messages, documents, and reports.

Request Manager

Is the administrator role for managing requests, such as viewing and handling all requests in the business unit and any applicable subbusiness units. Request managers handle both their own requests pending action and the requests pending action of other users. Request managers can search *all* requests in the Catalog system, while catalog users can search *only* their own requests.

Services Manager

Creates, defines, and manages services (not requests) for a specific tenant or business unit. This user also has administrative access to configure reports, dashboards, documents, and message alerts.

This role is most suitable when you want a user to create and maintain services. This user cannot request or subscribe to services.

This user can also handle requests pending actions, for example, by approving and rejecting requests.

Catalog Administrator

Creates, defines, and manages services for a specific tenant or business unit.

In addition, this user also has the same access rights as the request manager role.

This user can also request services but cannot subscribe to them.

Super Business Unit Administrator

Is the "root" user in a specific super tenant (super business unit). A super business unit is a business unit that contains one or more child business units. This administrator has *almost* complete access to the super business unit and all its sub business units. For example, anywhere in the super business unit, this administrator can create business units, create new users, and assign roles.

Service Delivery Administrator

Is the "root" user in the Service Provider (highest level) business unit. This user has complete system access to the root business unit and all sub business units.

Important! Assign this role carefully!

This role is available only for the Service Provider business unit, the default business unit created during installation.

Only this administrator has access to data mediation, system configuration, events, rules, and actions.

By default, at installation time, the Catalog system creates a user ID named *spadmin* with this role.

Default Role Specification

Service Delivery administrators can specify a [default role for all users](#) (see page 94).

Tasks that Each Role Can Perform

Each user (see page 77) can have different roles in different business units. The roles provide default access rights to various functions. In addition, administrators can use configuration settings to add access rights to a role or remove access rights from a role.

Note: For more details about the configuration settings, see the *Implementation Guide*.

Request-related functions are available through CA Service Catalog, while subscription and invoice-related functions are available through CA Service Accounting.

The following table lists the tasks that each role can perform. Explanations for the roles and authorization numbers appear after the table. The letter **X** indicates that the role *can* perform the task, while the dash (-) indicates that the user *cannot* perform the task.

Tasks	Roles							
	Cat Usr	Req Mgr	Cat Adm	End Usr	Adm	Svc Mgr	SBU Adm	SD Adm
Shopping								
By default, all users have all shopping functions, except as noted in Roles and Default Access Rights (see page 89). However, administrators can configure the access rights of each role to create proxy requests, edit requests, and so forth. All users can also delegate the use of their catalogs (see page 629) to create requests on their behalf.	X	X	X	X	X	-	X	X
Managing Requests								
View, edit, delete and cancel requests	X	X	X	X	X	X	X	X
Act on assigned requests pending action	X	X	X	X	X	X	X	X
Search for requests	X	X	X	X	X	X	X	X
View all items in a request	X	X	X	X	X	X	X	X
View request tracking and audit trail information	-	X	X	-	X	-	X	X
General								
View dashboards	X	X	X	X	X	X	X	X
Add personal dashboards	X	X	X	X	X	X	X	X

Task								
Create shared dashboards	-	-	-	-	X	-	X	X
View subscriptions and invoices	-	X	X	X	X	-	X	X
During checkout, change the Requested For user from the current setting to another account or user. That account or user requires a role in the business unit scope of the logged in user.	-	X	X	-	X	-	X	X
View and add News Messages	-	X	X	X	X	X	X	X
View Documents and View Reports	-	X	X	X	X	X	X	X
Managing the Catalog								
View and alter catalog services and service option groups	-	-	X	-	-	X	X	X
View and alter Service Builder configuration settings	-	-	X	-	-	-	X	X
Manage catalog entries or configuration	-	-	-	-	-	X	X	X
Manage subscriptions or invoices	-	-	-	-	X	-	X	X
Managing other elements								
Manage accounts within your business unit scope	-	-	-	-	X	-	X	X
Manage users with roles in your business unit scope	-	-	-	-	X	-	X	X
Manage the dashboard library for the business unit	-	-	-	-	X	-	X	X
Manage scheduled tasks	-	-	-	-	X	-	X	X
Manage reports	-	-	-	-	X	X	X	X
Manage Change Events and Alerts	-	-	-	-	X	-	X	X

Roles Key

Code	Role
Adm	Administrator
Cat Adm	Catalog Administrator
Cat Usr	Catalog User (none)
End Usr	end user
Req Mgr	request manager
Svc Mgr	service manager
SB Adm	Super Business Unit administrator

SD Adm   Service Delivery administrator

Tasks that Each Role Can Perform for Other Users

The following table displays the roles that can perform authorized tasks for themselves and for other accounts and users.

	Roles							
Can Perform Tasks for Themselves and	**Cat Usr**	**Req Mgr**	**Cat Adm**	**En d Usr**	**Ad m**	**Svc Mgr**	**SBU Adm**	**SD Adm**
Other accounts and users with roles in their business unit	-	X	X	-	X	X	X	X
Other accounts and users with roles in their business unit *and* any of its child business units.	-	X	X	-	-	X	X	X
Other accounts and users with roles in *all* business units, including *all* child business units	-	-	-	-	-	X	-	X

Default Role for All Users

The default role for all users (see page 77) applies to every user in the entire Catalog system. This default role applies to *all* users in *all* business units, including all child business units.

Only Service Delivery administrators can set this default role. They do so by logging in to the *root* (highest level) business unit and selecting Administration, Configuration, User Default Role.

The Catalog system automatically assigns this default role to every new user. However, administrators can optionally specify a different role for a user when they add (see page 81) or edit (see page 86) the user.

Relationship Between Users, Roles, and Login

Users (see page 77) and roles have the following relationship:

- A typical user belongs to one business unit.

- A user can optionally have different roles in different business units. For example, User A can have an end user role in the Finance business unit and a catalog administrator role in the IT business unit.

- If a user logs in to CA Service Catalog *without* specifying a business unit, the user receives the default role for all users (see page 94).

- If an integrating product (rather than CA Service Catalog) created the user, then the user is *not* assigned to a role or business unit. Instead, after the user logs in, the user receives the default role for all users (see page 94). Examples of integrating products include CA Service Desk Manager, CA Business Service Insight, and Reservation Manager.

Chapter 4: Managing Reports Using the Report Builder

This section contains the following topics:

Overview of Reporting Tasks

Using the Report Builder, you do the following to retrieve, format, and publish data in reports and other facilities:

- Add and edit data objects (see page 99), including these tasks:

    – Retrieve and display data from a data source

    – Specify dynamic selection criteria based on system variables or user input

    – Set up access to data objects according to role or business unit

- Add and edit data views (see page 107), including these tasks:

    – Create custom charts and tables, including three-dimensional (3D) ones

    – Provides summary level reports with drill-down capabilities to report details

    – Combine views from multiple data sources, text, and images in the report layout

- Add and edit layouts (see page 113), including these tasks:

  - Publish reports to the catalog (see page 115), where they can be requested or subscribed to by users or accounts

  - Publish reports to the dashboard library

  - Schedule reports to generate offline

  - Set up access to reports according to role or business unit

Note: Building reports requires knowledge of both the MDB schema and SQL syntax.

Note: You can also create reports using BusinessObjects Enterprise. For information about installing, configuring, and using BusinessObjects Enterprise, see the *Integration Guide*.

Structure of Reports

Reports consist of the following layers:

- The data object is the lowest layer. The data object defines the source of the data, the fields that make up each row of data, and the selection criteria.

  The data object produces a set of rows and columns of data that other facilities can use. These facilities include data views, form service option elements and runtime variables for other data objects.

  You can set permissions on a data object by business unit, functionality, and role.

- The data views define the format of a data object for a report. You can present the rows and columns of a data object in a table, chart, or both. You can highly customize tables and charts.

- The layout is a combination of data views, text, and images. The layout is best suited for presenting an overall view of business data.

  You can publish layouts as reports to the catalog. Users and accounts can request and subscribe to them. You can generate offline layouts which display a snapshot of data at a point in time.

  You can set permissions on a layout by business unit, functionality, and role.

CA Service Catalog includes several predefined data objects, data views, and layouts to use as-is or as models.

Data Objects

A data object defines the data to use for a chart or table. Data sources for a data object include the following:

- An ODBC-connected or JDBC-connected database management system (DBMS). Examples include an SQL Server or Oracle database.

- A delimiter-separated value file

- Any other data source that a Java report plug-in can access

You can perform the following tasks for data objects:

- View the list of data objects

- Organize them in folders

- Become familiar with the runtime variables (see page 100) in data objects

- Add a query runtime variable (see page 101) for use in a data object

- Add a data object (see page 102) or edit one

  Note: When you add or edit a data object, you can test it. You can also set permissions to specify the level of access for each role to the data object.

- Use predefined data objects (see page 105)

- Delete a data object

Runtime Variables

Data objects can use *runtime variables* to alter behavior and selection criteria dynamically.

For Query data objects, you can use runtime variables in the SQL query. As a prerequisite, define these runtime variables in the Variables list. That list includes default Query variables, and you can add your own.

For example, consider a report based on a SQL query that displays a list of users. The data object can take a Last_Name value as a runtime variable to indicate the starting characters of the last name. Users of the data object are prompted for the Last_Name value. You can use the runtime variable of type String named %Last_Name% in the SQL statement. Use this variable to restrict the results to user records that start with the value entered by the user. The following SQL statement provides a sample query:

SELECT first_name,middle_name,last_name FROM ca_contact WHERE (ca_contact.last_name like '%Last_Name%%')

Note: Users are prompted for only the runtime variables in the SQL query.

For plug-in data objects, you can pass runtime variables to the Java class as name-and-value pairs. As a prerequisite, verify that the Java report plug-in class takes the name-and-value pairs for the plug-in data object.

For example, consider the com.ca.usm.reporting.Plugins.RequestFulfillmentReport plug-in class. This plug-in class takes a parameter of Date type named START_DATE. Therefore, START_DATE is required in the data object that uses this plug-in class. In this case, do either of the following:

- Hard-code START_DATE as a constant

- Prompt the user for the date START_DATE value and pass that value to the plug-in

You can use contextual *system variables* with runtime variables, for the following purposes:

- As default values when you prompt the user

- As constant values to pass to the plug-in

- As values to use in a query

The contextual system variables are as follows:

Name	Variable
Current Day	%TODAY%
Previous Day	%TODAY%-Days(1)
Next Day	%TODAY%+Days(1)

First Day Of Month	%START_OF_CURRENT_MONTH%
Last Day of Month	%END_OF_CURRENT_MONTH%
First Day Of Year	%START_OF_CURRENT_YEAR%
Last Day Of Year	%END_OF_CURRENT_YEAR%
User Domain (Business Unit)	%USER_DOMAIN%
User ID	%USER_ID%

Add a Query Runtime Variable

You can add a custom runtime variable to use with a Query data object. An example is adding a query runtime variable that is a drop-down variable.

To add a Query runtime variable

1. Click Administration, Report Builder.

   The Report Builder page appears, with Data Objects selected in the left menu under the main menu.

2. Do *one* of the following:

   ■ Add a new data object

   ■ Edit an existing data object

   The page changes to match your action of creating or editing a data object.

3. Click Create Variable.

   The Create Runtime Variable or Enter Runtime Variable dialog appears.

4. Specify the name, type, and other data in the fields provided. Use the following guidelines:

   ■ To specify advanced data, click Advanced.

   ■ To view help text, click the Help (question mark) icon. It appears at the end of the title bar of the dialog.

5. Click Create Variable.

   The Catalog system saves your variable definition.

You can use the variable in an SQL query for a Query data object. Users are prompted to enter a value when they run the data object.

Note: When you add a query runtime variable that is a drop-down variable, the number of values in the resulting drop-down list is limited to 1000. If the report query returns more than 1000 values, the system truncates these additional values. Consequently, the user cannot view them in the drop-down list. If necessary, you can increase the number of values that appear in the drop-down list to be greater than 1000. For details, see the *Implementation Guide*.

Add a Data Object

You can add and edit data objects (see page 99) to define the source and content to retrieve for a report.

To add a new data object

1. Click Administration, Report Builder.

   The Report Builder page appears, with Data Objects selected in the left menu under the main menu. The existing data object folders appear.

2. Click Create Data Object.

   The Data Object Properties page appears.

3. Do the following:

   a. Select the Type of data object to create.

   b. (Optional) Select Show Advanced if you want to enter advanced specifications for the data object.

   c. Complete the fields, specifying the detailed options for the data object (see page 103). The exact fields that appear vary according to your selections for Type and Show Advanced.

4. Set the permissions for the data object, as follows:

   a. Click Permissions.

      The Set Permissions dialog appears.

   b. Set the access levels according to role, business unit, and CA EEM user groups, as applicable.

   c. Click OK.

      The Set Permissions dialog closes. You return to the Data Object Properties page.

5. Click *one* of the following:

   Save

      Saves the new data object without displaying any data.

   Save & Test

      Saves the new data object and displays the first 25 rows of the resulting data.

The Catalog system saves the new data object.

Note: You can also click Create Data View to create a data view from this data object.

Detailed Options for Data Objects

When you add or edit a data object (see page 102), you complete the fields, specifying the detailed options for the data object (see page 103). The exact fields that appear vary according to your selections for Type and Show Advanced on the page for adding or editing a data object. The following fields require explanation. These fields appear according to your selection for Type: Query, CSV File, or Plugin.

Query

Specifies the data to retrieve from a database query, as follows:

Database

Specifies the name of the database connection. The default is mdb.

Using the Query Builder, you can create a database connection definition that can use JDBC or an existing ODBC data source on the Service View server. For ODBC, administrators must define this data source identically on each Service View computer in your implementation. This requirement does not apply to JDBC; therefore, we recommend JDBC.

Table

Specifies a comma-separated list of tables for use in the SQL query. Use English characters *only*.

Fields

Specifies the names of the database table fields to produce as data object columns.

You can use the Search (magnifying glass) icon to populate these names automatically based on the query in the Query field. As a prerequisite, save the data object before you do so.

If you use an alias for a field in the query, you can use either name for the value of Fields. Use English characters *only*.

Query

Specifies the SQL query to use.

Optionally click Query Builder for help defining the query.

Note: You can use the Query Builder to define new ODBC or JDBC data sources.

Pivot, DB Locking, *and* **the Create Variables and Manage Variables buttons**

See Advanced Fields at the bottom of this topic.

CSV File

Specifies the data to retrieve from a file in delimiter-separated value format, as follows:

Fields

Specifies the names of the CSV file fields to produce as data object columns.

Optionally use the Find (magnifying glass) icon to populate the fields automatically, based on the contents of the CSV file named in the CSV File field.

You can use the Search (magnifying glass) icon to populate these names automatically based on the file name in the CSV File field.

CSV file

Specifies the path name of the CSV file. The path can be relative to %RPT_HOME% folder on the Service View server.

Note: If you use multiple Service View computers, this file must exist in the same folder on all of them.

Delimiter

Specifies the delimiter that separates the values in the file. Select a value from the drop-down list.

Pivot

See Advanced Fields at the bottom of this page.

Plugin

Specifies the data to retrieve from the output of a Java plug-in, as follows:

Note: For details on the available plug-ins, see the *Reference Guide*.

Enter the values for the fields shown for this type:

Fields

Specifies the names of the plug-in output fields to produce as data object columns.

Class Name

Specifies the full class name of the plug-in.

Arguments

Specifies the arguments for the plug-in.

Pivot *and* **DB Locking**

See Advanced Fields at the bottom of this page.

Advanced Fields

Selecting the Advanced check box exposes the following fields:

Pivot

Presents subtotals and counts of selected data.

Select Pivot to display the Select Pivot Fields dialog and configure the parameters.

This field applies to *all* selections for Type.

DB Locking

Specifies the type of database locking to use when reading from the database.

Select a value from the drop-down list.

For assistance, click the Help (question mark) icon next to this field.

This field applies when the Type is Query or Plugin.

Manage Variables and Create Variable buttons

Manage existing variables and add new ones. You can use these variables in SQL queries.

This field applies *only* when the Type is Query.

Pre-Defined Data Objects

CA Service Catalog supplied many predefined report data objects (data objects) that you can use in your reports. The following are the most commonly used predefined data objects. For details about these objects, see the comments supplied with them on the Administration, Report Builder, Data Objects menu.

Note: Data objects in this list that mention CA CMDB, CA Service Desk Manager, or CA APM are applicable *only* if the named product is integrated with CA Service Catalog. Otherwise, the data object returns irrelevant data. For information about the integrations with these products, see the *Integration Guide*.

Requests associated to assets

Returns the requests associated to assets in the business unit that you specify.

Requests associated to change orders and configuration items, by business unit

Returns CA Service Desk Manager change orders and CA CMDB configuration items associated to CA Service Catalog requests in the business unit you specify.

Requests associated to change orders and configuration items, by user

Returns CA Service Desk Manager change orders and CA CMDB configuration items associated to CA Service Catalog requests for the user ID that you specify.

Requests by business unit

Returns all requests for the business unit and date range that you specify.

Requests by status

Returns all requests for the status and date range that you specify.

Requests by year and month

Returns the total number of requests for each month in the year that you specify.

Request fulfillment

Returns the amount of time taken to approve, fulfill, and complete all requests within the date range that you specify.

Request item fulfillment

Returns the following:

- The time used to approve, fulfill, and complete the request for the specified service option element
- Request SLA violation data for each request

Request item fulfillment averages

Returns the following:

- The total number of requests that include the specified service option element
- The total time and the average time used to approve, fulfill, and complete the requested service option elements
- Request SLA violation data for each request item average

Request item fulfillment by request ID

Returns the following:

- The amount of time taken to approve, fulfill, and complete each service option of the specified request
- Request SLA violation data for each service option

Request SLA Instances

Returns all request SLA instances, with their SLA warning and violation thresholds.

Services associated to asset models

Returns the CA APM asset models associated to services and service options in the business unit that you specify.

Services associated to configuration items

Returns the CA CMDB configuration items associated to service in the business unit that you specify.

Total requests by business unit

Returns the total number of requests created per business unit for the date range that you specify.

Total requests by month

Returns the total number of requests grouped by month for the year that you specify.

Total requests by status

Returns the total number of requests grouped by status for the date range that you specify.

Note: CA Service Catalog also provides predefined data objects for CA Service Accounting, data mediation, financial reports, and metric events. For details, see the comments supplied with the related data objects.

Data Views

A data view formats the data produced by a data object. You can use data views to present data in tables, charts, or both.

You can perform the following tasks with data views:

- View the list of data views

- Assign data views to folders

- Add a data view (see page 108) or edit one

- Configure column rules (see page 108) when adding or editing data views

- Delete a data view

- Export a data view in PDF or CSV (delimited) format

- Manage off-line data views based on a data view

- Set permissions to specify the level of access for each role to a data view

Add a Data View

You use data views (see page 107) to define the presentation of a data object. Use column rules to configure how each field in the data object appears in a report.

To add a new data view

1. Click Administration, Report Builder.

   The Report Builder page appears, with Data Objects selected in the left menu under the main menu.

2. Click Data Views in the left menu.

   The list of existing data view folders appears.

3. Click Create Data View.

   The Data View page appears.

4. Complete the fields. Use the Choose Data Object icon to select a data object to use for this data view.

5. (Optional) Configure column rules (see page 108) for the data view.

6. Click Save.

   The Catalog system saves the data view.

You have added the new data view.

Configure Column Rules

You can use column rules to configure the presentation of a data view (see page 107) that appears as columns in a report.

To set up column rules for a data view

1. Click Administration, Report Builder.

   The Report Builder page appears, with Data Objects selected in the left menu under the main menu.

2. Click Data Views in the left menu.

   The list of existing data view folders appears.

3. Expand the folders to display the data view you want. Click the Edit (pencil) icon.

   The data view opens for editing.

4. Click the Column Rules button.

   The Create Column Rules dialog appears.

5. Select the column you want in the Settings for Column field.

   The settings appear for the selected column.

6. Complete the fields and configure the settings on each tab. For each tab, do the following, as needed:

   ■ Select Show Help to display help text for the tab.

   ■ Click Advanced to specify advanced settings.

   ■ Review the guidelines for completing the .

7. Click OK.

   The Catalog system saves the column rule.

You have set up the column rule.

Column Rule Tabs

When you configure column rules (see page 108) while adding a data view (see page 108), you optionally configure the specifications on each tab of the Column Rules dialog. Each tab contains one or more fields. For each tab, this topic explains the fields that require explanation.

Linking Tab

The following fields on this tab require explanation:

Link Items in Column *column-name* To Other Pages

Specifies whether to link the values in the column to another web page, either within or outside CA Service Catalog.

Select this field to activate it and to open the fields under it for editing.

When a user clicks a value in this column in a report, this link activates, and the linked item appears. Options follow:

Special

Specifies a link within CA Service Catalog to another data view, a report layout, or a GUI node.

Link Address

Specifies one of the following:

■ Another data view, a report layout, or a GUI node, if you used the Special link to add one.

■ A URL to a website or file share, if you did *not* use the Special field. Enter the URL manually.

Also, the Link Address field optionally specifies a variable from the data object on which the data view is based. Click the Insert Variables icon to add a variable to this field.

Formulas Tab

The following fields on this tab require explanation:

Apply Formula to Column *column-name*

Specifies whether to apply a JavaScript formula to the column value.

Select this field to activate it and to open the fields under it for editing.

Special

Specifies an image formula to apply on the data for the column. Use the formula to format a cell or row based on a value. Use the following format:

IMG:*image_file_path*

image_file_path specifies the folder and file name *under* the %USM_HOME%\view\webapps\usm folder.

For example, the following line displays the add.gif image in the cell for the column:

IMG:images/add.gif

To include the cell text also, enter the column name variable within the single quotes. An example follows:

IMG:images/add.gif]%Col1%

Note: If you use multiple Service View servers, verify that this file resides in the same folder location on all servers.

Column Formula

Specifies variable from the data object to use in the column formula. To use a variable, click the Insert Variables icon.

Examples of formulas using JavaScript follow:

100*%name%

Math.max(%Col1%,%Col2%)

(10*%Col1%)+(20/%Col2%))+' Units'

%name%'.toUpperCase()

Translations Tab

The following fields on this tab require explanation:

Apply Translations to Column *column-name*

Specifies whether to apply a translation formula to the column value.

Select this field to activate it and to open the fields under it for editing.

Translations

Substitutes a translated value for each data object value. You can optionally apply the translation before applying a formula to the column value.

For example, a column can return data as integer value *1* which means opened, and *2* which means closed. You can apply translations that do the following:

■ Display *Opened* in place of all values of type *1* in a column

■ Display *Closed* in place of all values of type *2*

Formatting Tab

The following fields on this tab require explanation:

Apply Formatting Settings to Column *column-name*

Applies formatting to a column. Examples include font, justification, highlighting, and color.

Select this field to activate it and to open the fields under it for editing.

Summary Tab

The following fields on this tab require explanation:

Summary

Adds summary information for the column.

Select this field to activate it and to open the fields under it so that you can select them. The fields you select appear in the summary of the report.

Layouts

You can use report layouts (layouts) to present multiple report elements as one report. You can design custom report layouts using text, images, URLs, other objects, and most importantly, data views. You can specify positions, sizes, colors, borders, styles, and other settings to obtain the desired layout for a report.

You can perform the following tasks with layouts:

- View the list of existing layouts

- Display a layout

- Organize layouts in folders

- Add a layout (see page 113) or edit one

  Note: When you add or edit a layout, you can set its status. You can also set the permissions to specify the level of access of each role to the layout.

- Manage off-line layouts based on a layout

- Publish a report to the catalog (see page 115)

Add a Layout

A layout defines the presentation of multiple report elements. Create and configure layouts to obtain the format you want for reports. You can use one of several sample layouts as a starting point. You can set permissions for the layout and optionally propagate those permissions to the underlying data views and data objects.

To add a new layout

1. Click Administration, Report Builder.

   The Report Builder page appears, with Data Objects selected in the left menu under the main menu.

2. Select Layouts in the left menu.

   The existing layout folders appear.

3. Click Create New Layout.

   The Edit Custom Report Layout page appears.

4. Click one of the following to add the new element you want to the layout:

- New Text

- New Data View

- New URL

- New Image

The new object appears in the layout background.

5. Drag the object to the location you want on the layout.

The object appears in the new location.

6. Configure the properties of the object by clicking the properties (i) icon and completing the fields.

The properties change to match your specifications.

7. Set the permissions for the layout, as follows:

a. Click Permissions.

The Set Permissions dialog appears.

b. Set the access levels according to role, business unit, and CA EEM user groups, as applicable.

c. Click OK.

The Set Permissions dialog closes. You return to the Edit Custom Report Layout page.

8. Click Save and specify a status, one of the following:

Created

Specifies that the layout is visible *only* to the user who created it (the creator). No other user is permitted to view the layout, regardless of the permissions set for it. This option enables the creator to complete it before anyone else can access it.

Available

Specifies that the layout is visible to the owner and to any user who is granted permissions to it.

9. Continue editing the layout as needed; save it periodically and when you are finished. Exit the Edit Custom Report Layout page by clicking Cancel.

You have added the layout.

Publish a Report to a Catalog

If you are using CA Service Catalog, you can publish a report to a catalog. After you do so, you can add the report to a service as a service option so that users can request the report.

Note: You can publish a report only if its status is *Available*.

Publish a report to a catalog

1. Click Administration, Report Builder.

    The Report Builder page appears, with Data Objects selected in the left menu under the main menu.

2. Select Layouts in the left menu.

    The existing layout folders appear.

3. Expand the folders by clicking the folder names or by clicking List View under Report Layout Objects.

    The list of reports appears.

4. Find the report you want and click the Publish Report to the Catalog icon in the Action column.

    The Catalog system adds the layout as a service option to the Published Reports service option group (see page 177).

You have published the report. You can now add the service option for the report to a service when you define the service (see page 164). As a result, users can request the report in the service.

View Published Reports

After you have [published a report](#) (see page 115), you and other users can view it.

Note: To view reports, you require access to the Reports menu option.

To view published reports

1. Click Home, Reports.

   The Reports page appears. It displays folders with reports under the General and Request tabs.

2. Click the General or Requested tab, as follows:

   General

   Contains the list of published reports that you can view.

   Requested

   Contains the list of published reports to which you have subscribed.

3. Expand the folder by clicking the folder names or by clicking List View under Report Layout Objects.

   The list of reports appears.

4. Find the report you want and display it by clicking the appropriate icon in the Action column.

You have viewed the published report.

Chapter 5: Managing Dashboards

This section contains the following topics:

Dashboards

A dashboard is a personal page containing elements from the dashboard library. These elements are named dash items. Users can create personal dashboards and access shared dashboards to which they have access. Administrators can create shared dashboards and manage the dashboard library.

Administrators and other users select the Administration, Dashboard menu to display and manage dashboards, to the extent permitted by their roles.

Dashboard Builder and Dashboard Library

You use the Dashboard Builder to manage the dashboard library. The dashboard library contains content that you can include in dashboards as dash items. To manage the dashboard library, you use the Administration, Dashboard Builder menu.

You can do the following tasks when managing dashboard library folders:

- Expand the top-level folders to display the child folders and their content elements (including their properties)

- View, copy, create, and delete folders and content elements in the Content Preview pane

- View and configure content elements (see page 121) the properties of folders and their contents in the Content Properties pane

- Administer dashboards (see page 118)

- Add a dashboard (see page 119)

- Publish content

Administer Dashboards

You can create and maintain dashboards (see page 117) to meet the needs of your organization.

Follow these steps:

1. Click Administration, Dashboard Builder.

    The Dashboard Library folders appear. They display the dashboards that you have permission to access.

2. Expand the library tree to display the category for which you want to administer dashboards.

3. Install ActiveX components, if you are prompted to do so.

    Note: Some dashboard items require ActiveX. When you first access a Dashboard Builder item, it prompts you to install ActiveX components in your browser, if necessary. In such cases, follow the prompts to install the ActiveX components. Afterwards, resume administering the dashboard.

4. Select the option you want from the Action drop-down list and click Go.

    Note: The options vary, according to the category that you have selected in the library tree.

5. Repeat these steps as needed for each dashboard that you must administer.

You have administered dashboards.

Add a Dashboard

You add personal or shared dashboards (see page 117) to provide expedited access to information and to frequently used features and functions of CA Service Catalog.

Follow these steps:

1. Click Home, Dashboards.

2. Click the << icon at the top right part of the page, and click Add Dashboard.

   The Dashboard Options page appears.

3. Name the dashboard and configure the other options, as follows. Click the Help (question mark) icon for assistance. The following fields require explanation:

   Shared Dashboard

   Creates a shared dashboard.

   Administrators use shared dashboards to publish information to users. If this option does not appear or if you do not select it, then this dashboard is available to you *only* (personal).

   Note: You can create a personal dashboard and share it later.

   When you select Shared Dashboard, several other fields appear. These fields are mutually exclusive. Select one of the following:

   ■   Accessible by Sub Business Units – Shares this dashboard with users in your business unit and its child business units.

   ■   Accessible by Role – Shares this dashboard with users who have the roles you specify. Only those roles can access the dashboard. If you do not specify your own role, you can no longer access the dashboard after you create it.

   Default Dashboard

   Sets the dashboard as the default dashboard.

   Full Screen

   Configures the dashboard to open in full-screen mode when users select it.

   Open in New Window

   Configures the dashboard to open in a new window when users select it.

   Disable Session Timeout

   Disables the session timeout feature. Consequently, users are not logged out if they are inactive longer than the session timeout value.

   Auto Arrange

   Arranges the dashboard items arranged automatically.

Lock Down Items

Fixes the location of the dash items so that other users cannot move them.

4. Click Add to create the dashboard.

    The new dashboard appears in the dashboard menu and is selected. The rest of the window is blank, because a new dashboard has no dash items.

5. Do the following to add dash items:

    a. Click Show Library from the Dashboard Administration menu.

        The Dashboard Library appears.

    b. Navigate the Library tree and locate the elements that you want to use on the dashboard.

    c. Drag the content elements to the place where you want them on the dashboard.

        The elements become the dash items.

    d. Adjust the size of the dash items as needed.

6. Set the properties of the dash items by clicking the Edit (pencil) icon on the dash item heading.

    Note: To delete a dash item from the dashboard, click the Delete (X) icon on the dash item heading.

7. Click Save Layout.

    The Catalog system saves the dashboard layout.

You have added the new dashboard.

Configure Content Elements

You configure content elements in <u>dashboards</u> (see page 117) to customize them to meet the needs of your organization.

Follow these steps:

1. Click Administration, Dashboard Builder.

   The Dashboard Library folders appear. They display the dashboards that you have permission to access.

2. Expand the folders and subfolders; display and select the content element you want.

   The details of the selected content element appear in the Content Preview and Content Properties panes.

3. Configure the fields on the Content Properties pane, and click Save.

   The following fields require explanation:

   Content Type

   Specifies the <u>type of content element</u> (see page 121).

   ACL Settings

   Specifies the access control list (ACL) settings.

   Use these settings to specify the level of access for each role to the content element.

   Note: Click the question mark help icon to display more information about the options on the Content Properties pane.

You have configured the content elements.

Types of Content Elements

When you <u>configure content elements</u> (see page 121) in the Content Properties pane of the Dashboard Library, you specify the type of content element. You can select from the following choices:

Folder

Configures the content element as a folder.

External Web Content

Configures the content element as a web page URL.

External XML Source

Configures the content element as an external web reference in XML format. If access to the XML content requires authentication, the web publishing framework automates the authentication login to gain access.

The web publishing framework can use Web Services to transform the information based on the needs of the user. You can view the XML directly. Alternatively, you can supply a custom XSL instead.

If you use XSL, you can embed it as part of the published data definition or obtain it through a URL.

Embedded HTML

Configures the content element to include the HTML to display.

You supply the information in HTML format. It is stored with the metadata in the Dashboard Library. Embedding HTML enables you to integrate with application data accessible through web controls and Java applets, such as Microsoft Outlook.

Embedded XML

Configures the content element to include the XML to display.

You supply the information in XML format. It is stored with the metadata in the Dashboard Library. Embedding XML enables you to view it directly. Alternatively, you can optionally supply a custom XSL instead of XML.

GUINode

Configures the content element to reference a CA Service Catalog page.

Specify the page as a Graphical User Interface (GUI) node.

GUINode XML

Configures the content element to reference internal CA Service Catalog data that you obtain through a GUI Node.

This setting enables you to define custom views of information without modifying back-end methods and customizing XSL style sheets.

Alternatively, you can optionally supply a custom XSL instead of XML.

Note: Access to the published data requires access to the GUI Node.

Managed Document

Configures the content element as a document maintained in CA Service Catalog.

You specify the path in the Filename field.

Note: This document must be in the same location on all Service View servers if you are using multiple Service View servers.

Chapter 6: Using Web Services

This section contains the following topics:

Overview

A web service is any collection of software operations or methods that is available over the Internet and that uses a standardized XML messaging system. The web services use XML to encode all communications: A client invokes a web service operation by sending an XML message and then waits for a corresponding XML response.

You can access the web services from any client that can use standard web service protocol. You can use the web services to automate business processes and reduce manual data entry.

The product includes several predefined web services covering a wide range of functions. For example, the UserService web service provides a *getUser* method and an *editUser* method that you can use to manage information about users. Moreover, the BusinessUnitService web service provides similar functionality for business units. This set of web services constitutes an application programming interface (API) for CA Service Catalog.

The web services use Simple Object Access Protocol (SOAP). SOAP is a lightweight, XML-based communication protocol and encoding format for inter-application communication.

The CA Service Catalog implementation of SOAP is Axis-compliant. You can access the web services from any Axis-compliant client. Implementers can use any programming language with which they are familiar to call exposed methods, using method call syntax. Implementers must be skilled users of the web services of the programming language they select.

The web services support the Web Service Description Language (WSDL). You can use WSDL to build stubs to access remote services. You can also use WSDL to export automatically machine-readable descriptions of CA Service Catalog deployed services from Axis.

View the API Documentation

You can view the CA Service Catalog web services API documentation for more information about how to use the web services. The API documentation is automatically generated Java documentation based on the CA Service Catalog web service methods.

Follow these steps:

1. Log in to CA Service Catalog.

    The CA Service Catalog main menu appears.

2. Click Administration, Tools, Links, Web Services API Documentation.

    The web services API documentation appears.

Note: You can also access the Java documents on the Service View computer by clicking Start, Programs, CA, Service Catalog, Documentation, Web Service API.

Deployment of Web Services

To make a web service accessible to a client application, deploy the web service (see page 126) and enable the method on the server. An authenticated user with proper permissions can deploy and undeploy services dynamically when the server starts.

By default, the Catalog system deploys all web services and their methods. Consequently, with the proper login credentials, a client application has access to all web services functionality. If you decide to undeploy a web service, you consider whether any client applications use it. For example, the CA Workflow process definitions and actors provided with CA Service Catalog use the web services heavily. Therefore, undeploying a web service can affect request approval and fulfillment business processes.

When a web service is deployed or undeployed, the Web Service Description Language (WSDL) file for the web service is updated. Each web service has its own WSDL that contains the information about currently deployed services, including method signatures. The URL format for the web service follows:

http://*hostname*:*port*/usm/services/*servicename*Service?wsdl

hostname:port

Specifies the Service View server name and port number.

servicename

Specifies the web service name.

For example, suppose you enter http://prod123:8080/usm/service/UserService?wsdl in your browser address field. The results display the WSDL contents for a Service View server named "prod123" running on port 8080. The results display all the web service methods and data structures for the User web services in XML format.

Note: You can also remotely invoke methods of the Axis server at runtime through an Axis-compatible client.

Deploy or Undeploy Web Services

To make a web service (see page 123) accessible to a client application, deploy the web service. When you do so, the service deploys dynamically when the server starts. Similarly, when you undeploy a web service, the service undeploys dynamically when the server starts. You can, for example, deploy web services that you previously undeployed or undeploy web services that you no longer use.

Follow these steps:

1. Click Administration, Tools, Links, Web Services Administration.

   The web service administration page appears. It lists the web services.

2. (Optional) Click the Plus sign next to the name of the web service.

   The service expands and displays the service method signatures.

3. Select the web service you want and click Deploy or Undeploy.

   Note: When you select a service, the right pane displays the details of the service including the input and output parameters for each method. You can click the plus sign next to the method name to view the details of the method.

You have deployed or undeployed web services.

How to Call Web Services

To call CA Service Catalog web services with a Java client, follow this process:

1. Verify that Java clients meet the prerequisites (see page 127).
2. Verify that the web services that you want to use are deployed (see page 125).
3. Generate the WSDL for each web service (see page 128).
4. Generate Java Stubs for each web service (see page 129).
5. Use the stubs to call each web service (see page 130).
6. Verify that you use login and logout methods (see page 130) efficiently.

7. Do one of the following:

   ■ If you are using Java programs, review the sample Java program to call a web service (see page 131). Use it as a model to call web services.

   ■ If you are using JavaScript, review the sample JavaScript program to call a web service (see page 135). Use it as a model to call web services.

8. See the following web sites for additional documentation:

   ■ If you are using Java programs, see the Axis and web service related-related sections of apache.org.

   ■ If you are using JavaScript, see the web service-related portions of the workshop sections of the Microsoft Developer Network (MSDN) on microsoft.com.

Prerequisites

Clients must meet the following prerequisites to call CA Service Catalog web services (see page 123):

■ Verify that the web services that you want to use are deployed (see page 126).

■ Axis clients require a Web Service Description Language (WSDL) file for initialization.

■ Verify that a WSDL file is generated for each web service (see page 128) when the Axis server is initialized. The WSDL file for a web service is updated each time the service is dynamically deployed or undeployed successfully.

   A WSDL file is required for each web service. Typically, when a web service is made available using Axis, a unique URL is associated with that web service. The URL name is typically http://localhost:8080/usm/services/*webservice*. An example is http://localhost:8080/usm/services/UserService for the web service named UserService.

   As a best practice, name your WSDL files using the convention *webservice*.wsdl, where *webservice* is the name of the web service. This document follows this convention.

■ Verify that each WSDL file contains the information, including method signatures, required to call currently deployed services. Using WSDL files enables you to differentiate the following parts of a service:

   – Abstract functionality description

   – Concrete details description, such as message format and communication protocol. Examples include SOAP, HTTP and MIME.

   Thus, you can reuse a WSDL file for different types of clients.

■ If you use a Java program to call the methods, verify that the following jar files are in the class path. These files are installed in %USM_HOME%\lib:

– axis.jar

– jaxrpc.jar

– commons-logging.jar

– commons-discovery.jar

– wsdl4j.jar

– mail.jar

Generate the WSDL File

Generating the Web Service Description Language (WSDL) file for each web service is required when you call web services with a Java client (see page 126).

To generate the WSDL file for a web service

1. Access the web service URL in a browser.

   You typically see a message referencing an Axis service and SOAP access.

2. Append **?wsdl** to the URL.

   Axis generates a service description for the deployed service and returns it as XML in the browser, for example:

   http://localhost:8080/usm/services/AccountService?wsdl

3. Save the output as a file named *webservice*.wsdl. Record the path name for future reference.

You have generated the WSDL. Use the webservice.wsdl file as input to web service calls (proxy-generation) when you generate java stubs for each web service (see page 129).

Generate Java Stubs

Generating Java stubs for each web service is required when you call web services with a Java client (see page 126).

To generate Java stubs for a web service

1. Verify that you are using the Axis WSDL2Java tool for generating the Java stubs.

    Doing so enables you to invoke the web service methods from your Java program.

    Note: You can obtain the Axis tools from the Apache Axis website (apache.org).

2. Open a command prompt and enter the command to generate Java stubs. Use the following command as a model.

    java org.apache.axis.wsdl.WSDL2Java -o . -ptesting.soap AccountService.wsdl

    This action generates the following files in the package named testing\soap.

    ■   *web_service_name*Impl.java

        This new interface file contains the appropriate java.rmi.Remote usages.

    ■   *web_service_name*ImplService.java

        This file is the Service Interface of the web service. The Service Locator implements this interface.

    ■   *web_service_name*ImplServiceLocator.java

        This file is the Helper factory to retrieve a handle to the service.

    ■   *web_service_name*SoapBindingStub.java

        This file is the client side stub code that encapsulates the client access.

    ■   Serialized beans generated dynamically

        These beans function as input and return types. They also catch exceptions.

3. Verify that the command generated files like the ones listed in the previous step.

You have generated Java stubs.

Use Java Stubs to Call Web Services

Using Java stubs to call web services is a required task when you call web services with a Java client (see page 126).

To use Java stubs to call web services

1. Verify that the stubs and bean classes that you want to use have been generated (see page 129).

2. Do *one* of the following:

   ■ If you are using Java programs, review the sample Java program to call a web service (see page 131). Use it as a model to call web services.

   ■ If you are using JavaScript, review the sample JavaScript program to call a web service (see page 135). Use it as a model to call web services.

You have used Java stubs to call web services.

How Clients Invoke Login and Logout Methods

When you call web services with a Java client (see page 126), a required process is clients invoking login and logout methods for each web service. A typical process follows:

1. The client uses a method for login and authentication.

   Each web service has a set of login methods. Client applications can use several login methods for authentication. For example, the *logIn* method takes the same parameters as the Login window: User ID, Password and Business Unit.

   To view the method parameter information, including signatures, use the following resources:

   ■ Web Services API documentation (see page 124)

   ■ SOAP administration interface for deploying or undeploying web services (see page 126)

2. The Catalog system authenticates the user and determines its role.

   Subsequent method calls operate within the scope of the access rights of the user, as if the user had accessed the GUI.

3. The client does the following:

   ■ States the service that it is calling

   ■ Provides the method name and its corresponding parameters before initiating the remote procedure call

   ■ Checks the WSDL file for this information if it is unknown

   Typically, the client already has this information.

4. The web service returns a session ID. This session ID is a required parameter that the client uses for the remaining web service calls. Because the underlying transport protocol can be either HTTP or non-HTTP, the authentication uses a common logIn web service.

   You can share the session ID across web services. For example, you can use the UserService logIn method to obtain a session ID. You can then use the session ID in a call to a Business Unit web service method.

5. This session ends when either of the following occur:

   ■ The client calls the logOut web service method.

      Once you are finished using a session ID, you call the *logOut* web service to terminate the session and invalidate the session ID. Managing sessions efficiently in this manner helps you obtain the best performance.

   ■ The client is idle for a period longer than the session timeout value.

      The client can use the session ID repeatedly within the timeout period. If the session times out, the session ID becomes invalid.

      To change the timeout value, update the administration configuration setting named User Default: Session Timeout.

      Note: For instructions to modify configuration settings, see the *Implementation Guide*.

How to Use a Java Program to Call a Web Service

Using a Java program to call each CA Service Catalog web service is a required task when you <u>call web services with a Java client</u> (see page 126).

1. Review the following Java-related specifications, which are used in this example. The specifications that you use in your calls vary according to your needs and goals.

   ■ A class named axisTester

      This class retrieves a JavaBean array containing the accounts in the business unit named Sales.

   ■ Server name: prod123

   ■ Port number: 8080

2. Review the following parameter-related specifications, which are used in this example. The specifications that you use in your calls vary according to your needs and goals.

- Web Service: AccountService

- URN: urn:usmAccountService

- Method: getAccountsOfBusinessUnit

- Input Parameters: sessionID, businessunitid

- Output Parameters: Array of Account beans

3. Review the sample programs:

-

-

4. Use the sample programs to create your programs.

Example for Retrieving a JavaBean Array

This example illustrates how to use a Java program to call a CA Service Catalog web service. In this example, the class named axisTester retrieves a JavaBean array containing the accounts in the business unit named CA. You can use this example as a model.

```
import java.io.*;

import java.util.*;

public class axisTester {

public static void main(String [] args) throws Exception {

try

{

// Construct the endpoint for calling the Account WebService
java.net.URL endpoint = new java.net.URL("http://prod123:8080/usm/services/AccountService");


// Call the Locator that implements the AccountService interface.
testing.soap.AccountServiceImplService service = new
testing.soap.AccountServiceImplServiceLocator();
// Use the service to get a Stub to the AccountService
testing.soap.AccountServiceSoapBindingStub fib =
(testing.soap.AccountServiceSoapBindingStub)service.getAccountService(endpoint);
```

```
// Log into the web service to get a session ID.
// loginID : The LoginID of the user(for example, spadmin).
// password : The User Password (for example, spadmin).
// bu : The business unit to which the user wants to log in (for example, CA)
// sessionID: handle to the session used for calling the other methods.
// The session ID can be used with any of the web services.
//String sessionID = fib.logIn(<loginID>,<password>,<domain>);
String sessionID = fib.logIn("spadmin","spadmin","CA");
// Using the sessionID the stub calls the method getAccountsOfBusinessUnit
// businessunitid: the ID for the business unit for which the accounts are to be retrieved (for example, "Sales")
//Account[] retAccounts = //fib.getAccountsOfBusinessUnit(sessionID,<businessunitid>);
Account[] retAccounts =
fib.getAccountsOfBusinessUnit(sessionID,"Sales");
for(int i = 0; i < retAccounts.length; i++)
{
System.out.println("value of AccountID = " + retAccounts[i].getAccountID());
System.out.println("value of AccountLabel = " + retAccounts[i].getAccountLabel());
}
// Log out. This terminates the session
fib.logOut(sessionID);
}catch(Exception e)
{
System.out.println("Throwing exception" + e.getMessage());
}
}
)
```

Example for Calling over an HTTPS Port

This example illustrates how to use a Java program to call a CA Service Catalog web service over an HTTPS port. You can use this example as a model.

Important! If Service View runs on the HTTPS port, use the keystore location and password in the call, as shown in this example.

```
package testing.soap;

public class axisTest {
  public static void main(String [] args) throws Exception {
    try
    {
      //   constructing the endpoint for calling the Account WebService
      java.net.URL endpoint= new java.net.URL("https://<hostname>:<https port
no>/usm/services/AccountService");

      System.setProperty("javax.net.ssl.trustStore", <keystore_location>);
      //  Eg:   <keystore_location> = //"C:\\j2sdk1.4.2_05\\jre\\lib\\security\\.keystore"
      System.setProperty("javax.net.ssl.trustPass",<keystore_password>);
      //  Eg:   <keystore_password> = "changeit"
      //calling the Locator which implements the Account WebService interface.
      testing.soap.AccountServiceImplService service =   new

testing.soap.AccountServiceImplServiceLocator();

      // Now use the service to get a Stub to the Account WebService

      testing.soap.AccountServiceSoapBindingStub fib =
(AccountServiceSoapBindingStub)service.getAccountService(endpoint);
          // The initial call for logging into the webservice.
      // loginID   : The LoginID of the user(Eg: spadmin).
      // password : The User Password.
      // domain     : The businessunit which the user wants to logIn to (Eg: ca.com.)This is optional and if the value
is null the default domain in taken.
      // sessionID: handle to the user used for calling the other methods.
      //              This is used for single sign on to other webservices.

        String sessionID = fib.logIn(<username>,<password>,<businessunit id>);

        // using the sessionID the stub calls the method getAccounts of BusinessUnit
      // businessunitid : the id of BU for which the user wants to extract the accounts //for.
      Account[] retAccounts = fib.getAccountsOfBusinessUnit(sessionID,<businessunit id>);
          for(int i =0; i< retAccounts.length;i++)
      {
        System.out.println("value of AccountID = " +retAccounts[i].getAccountID());
        System.out.println("value of AccountLabel = " +retAccounts[i].getAccountLabel());
          }
          // logging out: This erases the session of the user Logged In.
      fib.logOut(sessionID);
```

```
        }catch(Exception e)
    {
        System.out.println("Throwing exception" +e.getMessage());
```

Use a JavaScript Program to Call a Web Service

Developers can access web services directly through JavaScript programs. This ability enables web programmers and system administrators to invoke methods remotely through DHTML or Windows Scripting Host. The ability to call web services through client-side scripting gives web developers greater flexibility to create dynamic web sites.

To use a JavaScript program to call a web service

1. Review the following sample files and use them as models to create your own programs. These files reside in the directory named %USM_HOME%\ view\webapps\usm\admin.

   ■ soapTest_index.html

   ■ soapTest_bottom.html

   ■ soapTest.html, which contains the JavaScript code to call web service

   The sample files provide a sample HTML web page that gets the list of all accounts for a business unit, using synchronous method calls.

   Note: Because connections through HTTP and the Internet can be dropped, the sample files support both synchronous and asynchronous method calls. When you use asynchronous calls, the web browser does not lock during calls and responds to user input.

2. Open the soapTest_index.html file in your browser.

   The text of the file appears in your browser.

3. Complete the fields and run the file.

   The file runs and calls the web service. The page dynamically creates an HTML table with the list of accounts.

You have used JavaScript programs to call web services.

Addition of Attachments to Requests

The addRequestAttachmentWithPath web service replaces the earlier addRequestAttachment web service. The addRequestAttachmentWithPath web service does the following:

- Provides an automated, efficient, consistent, and reliable mechanism for adding attachments to requests
- Provides significant advantages over using earlier web services or manual methods

You can use each of the following methods to add attachments to requests with the addRequestAttachmentWithPath web service:

- Add attachments stored using the document management feature (see page 136) of CA Service Catalog
- Add attachments stored using third-party web-based distributed authoring and versioning (WEBDAV) (see page 138)

Important! Wherever applicable, replace instances of the old web service (addRequestAttachment) with the new web service (addRequestAttachmentWithPath).

Add Attachments Stored by the Product

You can optionally attach files to requests when the files are stored using the Document Management feature of CA Service Catalog.

Follow these steps:

1. Log in to CA Service Catalog.

2. Click Home, Documents, View Content.

   The View Content page appears.

3. Click Open as Web Folder. This link appears under the View Content tab.

   The Web Folder dialog appears.

4. Upload your documents, as follows:

   a. In the Document pane, select Upload Document in the Action drop-down list and click Go.

      The Add New Document dialog appears.

   b. Complete the fields on the dialog and click OK.

   Your documents are uploaded.

5. Call the web services that use addRequestAttachmentWithPath and pass the values you want to the input parameters.

    Use the following format for the parameter named attachmentPath:

    http://*username:password@computername*:port/usm/documents/*filename*

    username and **password**

    > Specifies the user name and password of the user who uploaded the documents. Alternatively, specifies a user with the Service Delivery administrator role.

    filename

    > Specifies the name of the file being uploaded, including its extension, if applicable.

You have attached files to requests when the files are stored using the Document Management feature.

Examples for attachmentPath Parameter

Some examples follow, for using the attachmentPath Parameter:

■ To upload a file to the Documents root folder *without* creating a folder, use the following format:

    http://spadmin:spadmin@computer-abc:8080/usm/documents/test.txt

■ To create a folder under the Documents root folder *and* upload a file to the new folder, use the following format:

    http://username:password@computer-def:port/usm/documents/*new folder*/filename

    new folder

    > Specifies the folder being created.

    filename

    > Specifies the file name of the attachment, including its extension.

■ To upload a file to a folder other than Documents root folder, use the following format:

    http://spadmin:spadmin@computer-xyz:8080/usm/documents/Web ServiceAttachments/test.txt

Add Attachments Stored by a WEBDAV

You can attach files to requests when the files are stored using a third-party web-based distributed authoring and versioning (WEBDAV). A sample WEBDAV is Microsoft Internet Information Server (IIS).

To add attachments stored using a WEBDAV

1. Pass the HTTP URL of the attachment.

2. Call the web services that use addRequestAttachmentWithPath and pass the appropriate values to the input parameter.

   Use the following format for the attachmentPath parameter:

   "http://*computername/websharefoldername/filename*"

   computername

   > Specifies the name of the computer from which you are adding the attachments.

   websharefoldername

   > Specifies the name of the new folder.

   filename

   > Specifies the file name of the attachment, including the extension for example, test.txt.

You have added attachments stored using a third-party WEBDAV.

Chapter 7: Using Service Builder

This section contains the following topics:

Service Builder

As an administrator, you use the Service Builder to create a service catalog to administer and provide access to services. You create and update entries in this catalog.

The service catalog consists of services, organized into a folder structure, available either to all business units or only a specific business unit. Services consist of one or more service option groups for IT services. Examples include hardware, software, new hire onboarding, and reservations for temporary physical or virtual resources. You can optionally specify charges (see page 390) when you design services.

You manage and configure services with the Service Builder according to the following categories:

- Services (see page 141)

  Add, change, and delete service and folders

- Service option groups (see page 169)

  Add, change, and delete the components of the services in the catalog

- Service option elements (see page 179)

  Add, change, and delete the details of these components

- Request SLAs and calendars (see page 253)

  Manage service level agreements for services

- Form Designer (see page 278)

  Create and maintain forms for use with services

- CA CMDB configuration item association

  Associate services with configuration items in CA CMDB

- Configuration

  Manage the Service Builder configuration settings

  Note: For details about setting the Service Builder configuration, see the *Implementation Guide*. For details about setting the CA CMDB configuration item association, see the *Integration Guide*.

Management of Services

The Services tab of the Service Builder page lists all services, arranged in folders hierarchically by category. Each business unit can have its own catalog of services. Alternatively, a business unit can use the catalog of a parent business unit, including the service provider (top-level) business unit.

Using the Service Builder, you can perform the following tasks to manage services:

- Change the business unit (see page 142)

- Basic operations

    - Add a new service (see page 142)

    - Add a new folder (see page 147)

    - Edit a folder or service (see page 151)

    - Delete a folder or service (see page 152)

    - Cancel a service (see page 153)

    - Copy a folder or service (see page 154)

    - Cut and paste a folder or service (see page 155)

- Advanced operations

    - Set service permissions (see page 156)

    - Set featured services (see page 157)

    - Service dependencies (see page 158)

    - Define service dependencies (see page 158)

    - Review and configure service and folder inheritance (see page 161)

    - Define a service (see page 164)

- Other advanced operations

    - Review thresholds for QoS SLAs (see page 165) and set them for request fulfillment reports (see page 166)

    - Use a service as a template (see page 167)

Change the Business Unit

By default, the catalog displayed is for the business unit for the role the user is logged in to.

To change the business unit

1. Click Service Builder, Services.

   The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Click Change Business Unit.

   The Search Business Unit window appears.

3. Use the Expand and Collapse icons to navigate the business unit tree or use the selection criteria and Search to locate the required business unit.

   Note: Only the business units permitted to be used by the user's role are included.

4. Click the business unit name in the tree to select it.

   The window closes and the catalog for that business unit is added to the display.

You have changed business units.

Add a New Service

A service consists of hardware, software, or other resources that users can request from the catalog. You create a service to group one or more service option groups together as a single service that users can request.

To add a new service

1. Click Service Builder, Services.

   The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the folder in which you want to add the new service. Click Add.

   Note: To add a service at the service provider (top) level, click the plus sign without expanding the tree.

   The Folder Details dialog appears.

3. Specify Service for the Type field, because you are adding a new service.

   The Folder Details dialog is set for a new *service* (rather than a new folder).

4. Complete the fields on the dialog (see page 143).

   Your specifications are recorded.

5. Click Set Permissions at the top of the dialog.

   The Set Permissions dialog appears.

6. Set access level controls for each role and click OK.

   The dialog closes. Each role receives the access rights that you specified to the new service.

7. Click OK.

   The service is saved.

You have added the service to the catalog. Define the service (see page 164) so that users can subscribe to the service or request it.

Folder Details Dialog for a New Service

When you add a new service (see page 142), the Folder Details dialog *for a new service* appears. You complete these fields as part of creating the service.

Adding a new folder (see page 147) is a related but unique task that displays a similar dialog, the Folder Details dialog *for a new folder*.

The following fields on the Folder Details dialog *for a new service* require explanation:

Date Created

Displays the date the service was created. This field is read-only.

Note: The date and time shown are based on the time zone on the Service View server. The same is true for the date fields that follow.

Date Available

Specifies the date when catalog users can request or subscribe to the service.

The dates and times that you enter in this field and the Date Unavailable field affect the availability of the service (see page 150) to catalog users.

Date Unavailable

Specifies the date when catalog users can *no longer* request or subscribe to the service.

Date Canceled

Specifies the cancellation date for all accounts with subscriptions to this service and all requests for this service. On this date, the applicable subscription and request statuses are set to Canceled status.

Selection Type

Specifies the number of times an account or user can subscribe to or request the service.

Select *one* of the following:

■   Not Selectable (no times)

■   Can Select Once (one time only)

■   Can Select Many Times (more than one time)

Approval Process

Specifies the approval process to use for the service when a user requests it.

Note: This Approval Process setting does *not* apply to subscriptions.

Select *one* of the following:

System approval process

Uses the following to determine whether the request for a service requires further approval:

■   The authorization level of the user requesting the service

■   The approval level of the service

If the authorization level of the user is less than the approval level of the service, the request requires further approval. In that case, the Catalog system does the following:

■   Assigns the manager of the user to approve the request

■   Places the request in the Requests Pending Action queue of that manager

This process repeats until the authorization level of the approver is equal to or higher than the approval level for the service.

Workflow driven approval process

Specifies that the Catalog system does the following:

■ Assigns the manager of the user to approve the request

■ Places the request in the Requests Pending Action queue of that manager

Note: Requests pending action can also be assigned to a group instead of an individual user. All users belonging to the assigned group see that request in their Requests Pending Action queues.

No approval

Approves the service automatically.

In policy driven request management (see page 451), you specify conditions in policies, based on the attributes of service options, services, requested items, users, and so forth. If a policy is active and a submitted request meets the condition in the policy, the following occurs: The users (assignees) named in the policy receive a request pending action to approve, reject, or fulfill a service option, service, or request. If a policy does not apply to a request, the Catalog system uses the approval process specified for the business unit in which the request was created: This setting is typically the workflow driven approval process.

Approval Level

Specifies the approver Authorization Level required for the service: Level 0, Level 10, Level 20, and so forth. Specify these values at your discretion, relative to the policies and needs of your organization.

Specify these values logically and consistently to help ensure proper approval operations in your organization.

Note: This setting applies *only* if you specify System approval process in the Approval Process field.

Status on Approval

Specifies the status of the request items in the service once the service is approved, *one* of the following:

■ Fulfilled

■ Pending Fulfillment.

Using Pending Fulfillment enables you to specify workflow processes that assign any additional tasks to fulfill the request, if necessary.

Note: This setting applies *only* if you specify System approval process or No approval process in the Approval Process field.

Version

Specifies the version number of the service.

Enter the value in the format of *major.minor.*

major

Specifies a value of 1000 or greater.

Note: The values from 0 to 999 are reserved.

minor

(Optional) Specifies any number.

Code

Specifies the text value to represent the product code, subscription code, SKU # or any other applicable value.

Sort By

Arranges the service option groups to appear to the user according to the category you select.

Categories include Name, Selection Type, Code, Date Created, and None. You can also select the category named Custom - use Sort Number. This option uses the Sort Number field in the service option group details.

Sort Number

Sorts the service according to the value you specify.

Note: This field applies *only* when the parent folder uses a value of Custom - use Sort Number in the Sort By field.

Image

Assigns an image to a service.

Click Browse to select the image from the %USM_HOME%\FileStore\images\offerings folder.

Important! The folder name *FileStore* is case sensitive. Therefore, use the correct case in path names and all other programmatic references.

Note: For information about setting up the filestore, see the *Implementation Guide*.

The recommended size for an image is no larger than 48 x 48 pixels. However, the size of an image for a featured service is fixed at 32 x 32 pixels. Consequently, the image is reduced or enlarged to 32 x 32 pixels, regardless of its original size.

Therefore, if you make this service a featured service (see page 157), verify that the image is legible when viewed at 32 x 32 pixels. Also, if you add images to several services in the same folder, verify that the sizes of the images are compatible. Verify that the images blend together to form a balanced layout that aligns with the Featured Item header.

Add a New Folder

Services can be contained in a folder structure. Folders can also be organized into a hierarchy of folders.

To add a new folder

1. Click Service Builder, Services.

    The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the folder under which you want to add the new folder. Click Add.

    Note: To add a new folder at the service provider (top) level, click the plus sign without expanding the tree.

    The Folder Details dialog appears.

3. Specify Folder for the Type field, because you are adding a new folder.

    The Folder Details dialog is set for a new *folder* (rather than a new service).

4. Complete the fields on the dialog (see page 148).

    Your specifications are recorded.

5. Click OK.

    The folder is saved.

You have added the folder to the catalog.

Folder Details dialog for a New Folder

When you add a new folder (see page 147), the Folder Details dialog *for a new folder* appears. You complete these fields as part of creating the folder. Adding a new service (see page 142) is a related but unique task that displays a similar dialog, the Folder Details dialog *for a new service*.

The following fields on the Folder Details dialog *for a new folder* require explanation:

Type

Specifies the type: Service or Folder.

Select Folder, because you are adding a new folder.

Date Created

Displays the date the folder was created. This field is read-only.

Note: The date and time shown are based on the time zone on the Service View server. The same is true for the date fields that follow.

The dates and times you enter in the date fields that follow affect the availability of the folder (see page 150) to catalog users.

Date Available

Specifies the date when catalog users can request or subscribe to the services in the folder.

Date Unavailable

Specifies the date when catalog users can *no longer* request or subscribe to the services in the folder.

Version

Specifies the version number of the service.

Enter the value in the format of *major.minor.*

major

Specifies a value of 1000 or greater.

Note: The values from 0 to 999 are reserved.

minor

(Optional) Specifies any number.

Code

Specifies the text value to represent the product code, subscription code, SKU # or any other applicable value.

Display Subfolders

Specifies whether the top-level subfolders under this catalog folder appear in the Browse section of the Requests page. Select *one* of the following options:

Use System Setting

Use the system or "global" setting to show or hide subfolders specified in the Browse Catalog: Show Subfolders parameter in the Request Management Configuration section of the Service Builder Configuration page.

If the value of the Browse Catalog: Show Subfolders parameter changes, all catalog folders that use the system setting are automatically updated accordingly.

Note: The Browse Catalog: Show Subfolders parameter and other Request Management Configuration settings are described in the *Implementation Guide*.

Show Subfolders

Specifies that this catalog folder *always* displays its subfolders in the Browse section of the Requests page, regardless of the value of the system setting.

Hide Subfolders

Specifies that this catalog folder *never* displays its subfolders in the Browse section of the Requests page, regardless of the value of the system setting.

Hiding these subfolders is often helpful when the number of them is so high that the following condition exists: Users must scroll through the Browse Catalog several times to view them all or to see the remaining catalog folders and their subfolders

Default: Use System Setting.

Sort By

Arranges the child folders or services according to the category you select.

Categories include Name, Selection Type, Code, Date Created, and so forth.

If you select Custom - use Sort Number, then the Catalog system uses the Sort Number field in the child service or folder.

The value of None provides no specific sort order.

Sort Number

Sorts the service according to the value you specify.

Note: This field applies *only* when the parent folder uses a setting of Custom - use Sort Number in the Sort Children By field.

Image

Assigns an image to a folder.

Click Browse to select the image from the %USM_HOME%\FileStore\images\offerings folder.

The maximum recommended size for a folder image is 48 x 48 pixels.

Important! The folder name *FileStore* is case sensitive. Therefore, use the correct case in path names and all other programmatic references.

Note: For information about setting up the filestore, see the *Implementation Guide*.

Availability of a Folder or Service

When you add a new service (see page 142) or add a new folder (see page 147), the values in the Date Available and Date Unavailable fields are important. These values affect when a folder or service appears in the catalog for users to select, as follows:

- The date and time for the user setting these values can be the same (or earlier) date and time as the Service View server. In that case, the change takes place immediately.

- The date and time for the user setting these values can be *later than* the date and time on the Service View server. In that case, the availability of the folder or service is not affected until Service View runs its catalog management logic. This logic typically runs at midnight but also runs when the Service View service is restarted.

Edit a Folder or Service

You can edit the values for an existing folder or service when you update them for any reason.

To edit a folder or service

1. Click Service Builder, Services.

    The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the folder or service that you want to edit.

    Note: To add a new folder at the service provider (top) level, click the plus sign without expanding the tree.

3. Click the pencil edit icon in the Actions column for the folder or service you want to edit.

    Note: To edit the root folder of a catalog, click Edit in the heading bar for the catalog.

    The Folder Details dialog appears.

4. Edit the fields, as follows:

    ■   If you are editing a service, edit the same fields as when you <u>add a new service</u> (see page 142).

    ■   If you are editing a folder, edit the same fields as when you <u>add a new folder</u> (see page 147).

5. Click OK.

    The Catalog system saves your updates.

You have edited the folder or service.

Delete a Folder or Service

You can delete an existing folder (and its child folders and services) or service.

Important! Deletion is permanent. Deleted folders and services cannot be recovered.

To delete a folder or service

1. Click Service Builder, Services.

   The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the folder or service that you want to delete.

   Note: To add a new folder at the service provider (top) level, click the plus sign without expanding the tree.

3. Do one of the following *if* the folder or service that you want to delete contains inherited folders or services. Otherwise, skip to the next step.

   ■ Delete the child, inherited folder or service

   ■ Break the inheritance by editing (see page 151) the child, inherited folder or service

   Note: You cannot delete a folder or service that contains inherited folders or services.

4. Click the Delete icon in the Actions column for the folder or service you want to delete.

   The folder or service is deleted.

You have deleted the folder or service. Deleting services affects the subscriptions and requests (see page 152) based on the service.

Effects of Deleting a Service

Deleting folders or services (see page 152) affects the subscriptions and requests for the deleted services, as follows:

Subscriptions

The account subscribed to the service being deleted no longer contains or lists that service.

Requests

The status of requested service options for the account change based on their original status, as follows:

Original Status	New Status
Not Submitted	The service is deleted from the request

Submitted, an approval status, a fulfillment status, Pending Resource Assignment, or Resource Assigned	The service and its service options are set to Cancelled
Completed	The default cancellation status (Pending Cancellation or Cancel) if CA Service Accounting is installed; Cancelled if CA Service Accounting is *not* installed
Pending Cancellation or Cancelled	same as original status

Cancel a Service

You can cancel a service for several reasons. For example, the service is no longer needed, or it was incorrectly assigned.

To cancel a service

1. Click Service Builder, Services.

    The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the service that you want to edit.

3. Click the Edit icon in the Actions column for the service you want to cancel.

    The Service Details dialog appears.

4. Specify a Date Canceled and click OK.

    The Catalog system saves your changes.

You have canceled the service. Canceling a service has the <u>same effects as deleting a service</u> on the subscriptions and requests based on the service. These effects are activated when Service View runs its catalog management logic. This logic typically runs at midnight but also runs when the Service View service is restarted.

Copy a Folder or Service

In the catalog, you can copy services and folders. When you copy a folder, you also copy its child folders and services. Copying services and folders can be helpful in several circumstances. Administrators can, for example, create new services by copying and modifying existing services. They can also copy an established service from one business unit to another. This topic explains how to copy *without* inheritance. To use inheritance, you create an inherited folder (see page 162) or create an inherited service (see page 163).

To copy a folder or service

1. Click Service Builder, Services.

   The Services page appears and displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the folder or service that you want to copy.

3. Click the Copy icon in the Actions column for the folder or service you want to copy.

   A message appears asking which *one* of the following you want to include in the copy:

   ■   All associated service option groups

   ■   Only links to these service option groups

4. Answer the prompt.

5. Expand the tree and select the folder to which you want to paste the clipboard contents.

6. Click the Paste icon in the Actions column for that folder.

   The folder (and its child folders and services) or service is pasted to the new location.

   Note: To paste the clipboard contents to the root folder of the catalog, select the header bar for the catalog and click Paste.

You have copied the folder or service.

Cut and Paste a Folder or Service

In the catalog, you can cut and paste an existing folder (and its child folders and services) or service. Doing so is helpful when you must move the folder or service from its existing location to a new location.

To cut and paste a folder or service

1. Click Service Builder, Services.

   The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the folder or service that you want to copy.

3. Click the Cut icon in the Actions column for the folder or service you want to cut.

   The folder (and its child folders and services) or service is cut and moved to the clipboard.

4. Expand the tree and select the folder to which you want to paste the clipboard contents.

5. Click the Paste icon in the Actions column for that folder.

   The folder (and its child folders and services) or service is pasted to the new location. The folder or service is also deleted from original location.

   Note: To paste the clipboard contents to the root folder of the catalog, select the header bar for the catalog and click Paste.

You have cut and pasted a folder or service.

Set Service Permissions

Setting permissions for a service means specifying the level of access that each role has to the service. Setting permissions helps safeguard the security and integrity of services.

To set permissions for services

1. Click Service Builder, Services.

   The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Expand the tree and select the service whose permissions you want to set.

3. Click the Edit button.

   The Service Details dialog appears.

4. Click the Set Permissions button at the top of the dialog.

   The Set Permissions dialog appears. It displays access control level settings for each role and group (if applicable).

5. Select Grant All Access to Everyone or set the Request/Subscribe permissions by user role and group (if applicable).

6. Exit the dialogs by clicking OK on each of them.

   The Catalog system saves your updates.

You have set permissions for the service.

Set Featured Services

As an administrator, you can specify services to appear to catalog user as featured services, both in the catalog root folder and its subfolders. Specifying featured services is helpful to highlight popular or required services so that users can more easily find and request them.

To add a service to the featured items list for an existing folder or service

1. Click Service Builder, Services.

   The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Do one of the following:

   ■ To define featured services for the catalog root folder *only,* click the Feature *button* at the top of the Catalog bar.

   ■ To define featured services for another folder or service, do the following:

      a. Expand the tree and select the folder or service that you want to feature.

      b. Click the Feature *icon* under Actions for the selected folder or service.

   The Feature Items window opens.

3. Expand the Catalog Tree on the left side of the Feature Items window. Select the services to appear as Featured Items for the folder or service being edited.

4. Click the arrow to add the selected services to the Features Services list.

   Note: The Inherited Services portion of the Feature Items window displays the services that are already featured with this folder or service. They are featured with a parent folder whose Feature with Children check box is selected.

5. Select each service in the Featured Services list for folders and specify the following options:

   Feature with Children

      Indicates that this service is featured when the user displays subfolders.

   Sort By

      Specifies how to sort the services in the Featured Items section of the Requests window when the user selects the folder.

   Sort Order

      Specifies how to sort the services in the Featured Items section of the Requests window when the user selects the folder.

6. Verify that any image associated with this service is legible when viewed at 32 x 32 pixels. The guidelines for images are the same as when you add a new service (see page 142).

7. Click Save and click Close.

    The Featured Items window closes. You return to the Service Builder, Services page.

You have set featured services.

Service Dependencies

A dependency in the catalog allows a service state to change depending on the state of another service. You can also set the availability of a service to a user based on the business unit of the user that is viewing the catalog.

The ability to define dependencies of services allows you to define how one service behaves based on how you select or clear another service. For example, if you are subscribing to Service A, the following may happen to Service B:

■   Service B may become disabled. The selection check box is grayed out and cannot be selected or cleared.

■   Service B may become subscribed to. The check box displays selection whether it is enabled or disabled.

■   Service B may become unsubscribed to. The check box is cleared whether it is enabled or disabled.

Define Service Dependencies

You can define dependencies between services, based on user selection. Doing so helps ensure that the Catalog system automatically includes or excludes subscriptions or disables services according to your specifications.

To define service dependencies

1. Select Service Builder, Services.

    The Services page appears. It displays the folders containing services, organized hierarchically according to category.

2. Click Change Business Unit (see page 142), if necessary, and select the business unit whose service dependencies you want to define.

    The catalog for the business unit appears.

3. Click Define Dependencies on the Catalog bar.

    The Define Dependencies page appears.

4. Select one of the Type of Action choices:

    Business Unit to Service Assignment

    Specifies which services appear in the catalog of the child business unit.

    When you select this option, a list of child business units appears. When you select a business unit, the list of available services appears. You specify whether to include the service in the catalog of the child business unit.

    Service to Business Unit Assignment

    Specifies which services appear in the catalog of the child business unit.

    When you select this option, a list of services appears. When you select a service, the list of child business units appears. You specify whether to include the service in the catalog of the child business unit.

    Subscribe Services

    Specifies which services are included automatically in a subscription to another service. When a user subscribes to the selected service, the Catalog system *also* subscribes the user to the service or services that you specify.

    When you select this option, a list of services appears. When you select a service, the list of remaining services appears. You specify which of the remaining services (if any) to include automatically when a user subscribes to the selected service.

    Un-subscribe Services

    Specifies which subscribed services are canceled automatically with the cancellation of another service. When a user cancels the selected service, the Catalog system *also* cancels the subscription (of this user) to the service or services that you specify.

    When you select this option, a list of services appears. When you select a service, the list of remaining services appears. You specify which of the remaining services (if any) whose subscriptions cancel automatically when a user cancels a subscription to the selected service.

Disable Services

Specifies which services are made *unavailable* (disabled) automatically as a result of a subscription to another service. When a user subscribes to the selected service, the Catalog system *also* disables (for this user) the service or services that you specify.

When you select this option, a list of services appears. When you select a service, the list of remaining services appears. You specify which of the remaining services (if any) to disable automatically when a user subscribes to the selected service.

5. Click Save Dependency.

The Catalog system saves your specifications.

6. Repeat the previous steps until you are finished specifying dependencies.

7. Click the Close icon.

The Define Dependencies dialog closes.

You have defined dependencies among services.

Service and Folder Inheritance

You can configure the inheritance folders and services, as follows:

- Inherit folders and services from the catalog of one business unit to the catalog of another business unit.

- Inherit folders and services from one part of the catalog hierarchy to another.

- Create an inherited (child) folder (see page 162)

- Create an inherited (child) service (see page 163)

After you create the inherited folder or service, the following rules apply:

- New folders and services that are added to the parent folder are *not* added to the child folder. You cannot change the service option group definitions of the services in the child folder structure. However, you can add new folders and services.

- Over time, you update the services (including the service option groups) in a parent folder. Your changes are reflected in the services of its child folder.

- Similarly, over time, you update the parent services. These updates can include the service option groups, default definitions, and so forth). Such changes are reflected in the child services of the parent services.

- You can change the details of the child folder. However, if you change the parent folder details, your updates automatically take effect in the child folder. Your updates in the parent folder override any existing values in the child folder. The only exceptions are as follows:

  - For folders, the folder name is preserved.

  - For services, the service name, dates, and status are preserved.

- If the service option group definition of the new, dependent service is changed, the inheritance relationship with the parent service is removed. Consequently, changes to the parent service are no longer reflected in the child service.

You can also create an inherited service option group (see page 176).

Create an Inherited Folder

You can create an inherited folder by copying an existing folder and using inheritance. The inherited (child) folder contains the same contents as its parent folder. When the parent folder is updated, the same updates occur in the child folder automatically. If you do not want to use inheritance, you can copy a folder or service *without* inheritance (see page 154).

Follow these steps:

1. Review the preliminary information about service and folder inheritance (see page 161).

2. Click Service Builder, Services.

   The Services page appears and displays the folders containing services, organized hierarchically according to category.

3. Expand the tree and select the folder that you want to copy.

4. Click the Copy icon in the Actions column for the folder or service you want to copy.

   A message appears asking which *one* of the following you want to include in the copy:

   ■ All associated service option groups

   ■ Only links to these service option groups

5. Answer the prompt.

6. Expand the tree and select the folder where you want to paste the inherited folder.

7. Click Paste as Inherited.

   The new child folder is created at the *root* level of the hierarchy. This folder has the same characteristics as the parent folder that was in the clipboard. However, the names of this folder and any child folders and services use the prefix "Inherited from."

   Note: To add the inherited folder under an existing folder, click the Paste as Inherited icon in the Action column of the target folder.

You have created the inherited folder.

Create an Inherited Service

You can create an inherited service by copying an existing service and using inheritance. The inherited (child) service contains the same contents as its parent service. When the parent service is updated, the same updates occur in the child service automatically. If you do not want to use inheritance, you can copy a folder or service *without* inheritance (see page 154).

Follow these steps:

1. Review the preliminary information about service and folder inheritance (see page 161).

2. Click Service Builder, Services.

   The Services page appears and displays the folders containing services, organized hierarchically according to category.

3. Expand the tree and select the service that you want to copy.

4. Click the Copy icon in the Actions column for the folder or service you want to copy.

   A message appears asking which *one* of the following you want to include in the copy:

   ■ All associated service option groups

   ■ Only links to these service option groups

5. Answer the prompt.

6. Expand the tree and select the folder to which you want to paste the clipboard contents.

7. Click Paste as Inherited.

   The new child service is created at the *root* level of the hierarchy. This service has the same characteristics as the parent service that was in the clipboard. However, the names of this service and any child folders and services use the prefix "Inherited from."

   Note: To add the inherited service under an existing folder, click the Paste as Inherited in the Action column of the target folder.

You have created the inherited service.

Define a Service

After you create a service, define it. *Defining* a service includes associating it with one or more service option groups. When catalog users view the service, they can verify whether to request it based on the service option group it contains.

To define a service

1. Select Service Builder, Services.

    The services page appears.

2. Expand the tree to display the service of interest and click the Define icon.

    The page for defining that service appears, with the name of the service at the top of the page. This page appears in *one* of the following formats:

    ■   If no service option groups are associated with this service, all service option groups appear. The Show All link is pre-selected.

    ■   If one or more service option groups are already associated with this service, those service option groups appear. The Show Selected link is pre-selected.

3. Click the Show All link, if it is not already selected.

    The list of all service option groups appears (or is refreshed).

4. Do all of the following that apply:

    ■   Include a service option group in the service by selecting its check box in the Include column.

    ■   Display and update a service option group by clicking the Plus sign next to it.

        The details of the service option group appear. Do all of the following that apply:

        –   Set permissions on which user roles can subscribe to or request the selected service option. To do so, select the service option and click the Set Permissions icon in its Actions column.

        –   [Set thresholds for QoS SLAs for request fulfillment reports](#) (see page 166). To do so, select the service option and click the Set SLAs icon in its Actions column.

        –   Select the service options that are included in the service by default. To do so, select the check box for the service option in the Default column.

        –   Select the service options that are required in the service. To do so, select the check box for the service option in the Include column.

    ■   Add a new service option group to the list of all service option groups by clicking Add New Service Option Group.

    ■   Set permissions on which user roles can subscribe to or request the service option group. To do so, click the Set Permissions icon in the Actions column for the service option group.

■ Delete a service option group from a service by clearing the check box for the service option group in the Include column.

■ Delete a service option group from the Catalog system by clicking the Delete icon for the service option group in the Actions column.

5. Click Save Selection.

Your changes are saved.

You have defined a service.

Thresholds for QoS SLAs

Note: Request service level agreements (SLAs) are a feature of CA Service Catalog, while Quality of Service (QoS) SLAs are available *only* if CA Service Catalog is integrated with CA Business Service Insight. The terms *request SLA* and *QoS SLA* are used when needed in the documentation to distinguish between the two types of SLAs.

You can establish thresholds for QoS SLAs for use with Request Fulfillment reports. An SLA threshold specifies the following:

■ Starting status

■ Ending status

■ Duration (days, hours and minutes) for warning status

■ Duration for a violation status

If the request for the service option takes longer than the SLA duration value to move from the starting status to the ending status, then the request reaches a warning or violation status.

The Request Fulfillment reports use the SLA threshold specification mentioned earlier. By default, the report includes the following phases:

■ The approval phase

■ The fulfillment phase

■ The combination of approval and fulfillment phases

The following table shows summary data for these items:

Starting Status	Numeric Value	Ending Status	Numeric Value
Submitted	200	Approval Done	999
Approval Done	999	Completed	2
Submitted	200	Completed	2

You can also configure the reports to display the following:

■ The time it takes for the status of a service option to move from the starting to the ending phase

■ The average time in each phase

■ The violation time for the SLA phases

■ The color code of warnings and violations for the service options for which SLA thresholds have been set

If you report on different SLA thresholds, alter the STATUS_RANGES values for the Request Fulfillment reports.

Note: The SLA violation duration for the Submitted to Completed phase is used for the "estimated time to fulfill" value. This value appears in the catalog in the "fulfillment details" for a service option.

Set Thresholds for QoS SLAs

Note: Request service level agreements (SLAs) are a feature of CA Service Catalog, while Quality of Service (QoS) SLAs are available *only* if CA Service Catalog is integrated with CA Business Service Insight. The terms *request SLA* and *QoS SLA* are used when needed in the documentation to distinguish between the two types of SLAs.

You can establish thresholds for QoS SLAs (see page 165) for use with Request Fulfillment reports. Doing so is an optional task when you define a service (see page 164).

Note: The SLA violation duration for the Submitted to Completed phase is used for the "estimated time to fulfill" value. This value appears in the catalog in the "fulfillment details" for a service option.

To set SLA thresholds for Request Fulfillment reports

1. Select Service Builder, Services.

   The services page appears.

2. Expand the tree, locate the service of interest, and click its Define icon in the Actions column.

   The page for defining that service appears, with the name of the service at the top of the page. This page appears in *one* of the following formats:

   ■ If no service option groups are associated with this service, all service option groups appear. The Show All link is pre-selected.

   ■ If one or more service option groups are already associated with this service, those service option groups appear. The Show Selected link is pre-selected.

3. Click the Show All link, if it is not already selected.

   The list of all service option groups appears (or is refreshed).

4. Expand the service option group that contains the service option of interest.

   The details for the service option group appear. The service options in the group appear.

5. Select the service option you want and click the Set SLAs icon in its Actions column.

   The SLA Definition dialog for the service option appears.

6. Click Add.

   The new SLA is added. A new SLA row appears.

7. Specify the following and click OK:

   ■ The starting status

   ■ The ending status

   ■ The warning and violation thresholds

   The Catalog system saves your changes.

You have set SLA thresholds for a service option for the Request Fulfillment reports.

How to Use a Service as a Template

You can optionally create a service to use as a template for creating all services in your catalog (a template service). Creating a template service helps ensure that all services in your catalog have a consistent look-and-feel. To create a template service, follow this process:

1. Add a new service (see page 142), using the following guidelines:

   ■ Include the word "template" in the name.

   ■ Mark the service as Unavailable by setting the unavailable date to the current date.

   ■ Select the service option groups that you want to be in the template service.

2. Inform other administrators about the template service. Tell them to follow the remaining steps to use it to add a new service:

   a. Copy and paste the template service to an appropriate "holding" folder.

   b. Rename the service each time that you copy it to a different folder.

   c. Add new service option groups to the service, if necessary.

   d. Remove existing service option groups from the service, if necessary.

3. Do *one* of the following to modify a service option group in a service created from your template service:

    ■   To apply your changes to *all* services that use the service option group, <u>edit the service option group definition</u> (see page 174) directly.

    ■   To apply your changes *only* to the new service that you are creating from the template service, do the following:

        a. Copy and rename the service option group.

        b. Modify the new service option group as required. For example, update its name, add one or more new service option elements to it, or delete one or more service option elements from it.

        c. Add the new service option group to your new service.

        d. Remove the original service option group from the service.

    Note: When you copy a service, the original service option group or groups are included, *not* copied. Consequently, your changes to a service option group apply to *all* services that include it, not only to the new service that you are adding. Sample updates include changing its details and adding or removing service option elements. Thus, when you modify a service option group, you are updating the *original* service option group, *not* a copy of it.

4. <u>Define a service option element</u> (see page 179) in a service option group, or update it, if necessary.

    Your updates apply to *every* service option group--and consequently *every* service--that includes the service option element.

Service Option Groups

Each service in the catalog consists of one or more service option groups. Service option groups are bundled groups of individual service option elements. These elements define the characteristics of the catalog items. Each business unit can have its own set of service option groups.

Note: To display the service option groups for an additional business unit, click Change Business Unit.

You can perform the following *basic* tasks to manage service option elements:

- View a service option group definition (see page 170)

- Add a service option group (see page 171)

- Use a tiered service option group (see page 173)

- Review the availability of a service option group (see page 172)

- Edit a service option group (see page 174)

- Copy and paste service option groups

- Delete a service option group (see page 174)

You can perform the following *advanced* tasks to manage service option elements:

- Use inheritance to add a service option group (see page 176)

- View service option group dependencies (see page 177)

- Review published reports (see page 177) and publish a report layout (see page 178) for a service option group

- Define a service option group (see page 178)

- Associate a CA APM model

  If your CA Service Catalog implementation integrates with CA APM, you can associate one or more CA APM models with a service option.

  Note: For more information about associating CA APM models with a service option, see the *Integration Guide*.

View a Service Option Group Definition

A service option group definition is a display-only view of the service option elements in a service option group.

To view a service option group definition

1. Select Service Builder, Service Option Groups.

   The Service Option Groups page appears.

2. Click the option button next to the service option group whose definition you want to view.

   The service option group is selected.

3. Click the View Definition link at the top of the business unit service option group list.

   The read-only Service Option Group Definition dialog appears.

You have viewed the service option group definition.

Add a New Service Option Group

You can add a new service option group (see page 169) when required. Each service that users request from the catalog contains one or more service option groups. You can use the same service option group in multiple services.

To add a new service option group

1. Click Service Builder, Service Option Groups.

    The Service Option Groups page appears.

2. Click Add.

    The Service Option Group Details page appears.

3. Complete the fields.

    The following fields may require explanation:

    Date Created

    Displays the date the service option group was created. This field is read-only.

    Note: The date and time shown are based on the time zone on the Service View server. The same is true for the date fields that follow.

    Date Available

    Specifies the date when this service option group becomes available for catalog administrators to include in a service when they define the service.

    The dates and times you enter in this field and the Date Unavailable field affect the availability of a service option group (see page 172) to catalog administrators.

    Date Unavailable

    Specifies the date when catalog users can *no longer* request or subscribe to the service option group.

    Type

    Specifies the type of service option group. This value determines how you can include this service option group within other service option groups as a characteristic of a service option element. Select one of the following:

    ■   Fixed - A fixed service option group presumes a fixed cost or a variable cost based on usage

    ■   Tiered - A tiered service option group presumes a variable cost value.

    Default: Fixed

Selection Type

Specifies how catalog users select the options in the service option group while requesting a service that includes the service option group:

■   Automatically Chosen – All service options in the service option group are included automatically. They are all required, and the user cannot remove them.

■   Choose Exactly One – Users can select only one service option, by clicking an option button.

■   Choose Many – Users can select one or more service options, by selecting check boxes.

Default: Automatically Chosen

Code

Specifies a text value for a code of your choice. Examples include the product code, subscription code, and SKU number.

Sort Number

Specifies how to sort this service option group when it is included in a service that uses a Sort By setting of Custom - use Sort Number.

Description

Specifies a description of the service option group for catalog administrators *only*. This value appears in the Service Builder but does *not* appear in the catalog for users.

4.   Click OK.

You have added the new service option group.

Availability of a Service Option Group

When you <u>add a new service option group</u> (see page 171), the values in the Date Available and Date Unavailable fields are important. These values affect when a folder or service appears in the catalog for users to select, as follows:

■   The date and time for the user setting these values can be the same (or earlier) date and time as the Service View server. In that case, the change takes place immediately.

■   Conversely, the date and time for the user setting these values can be *later than* the date and time on the Service View server. In that case, the availability of the folder or service is not affected until Service View runs its catalog management logic. This logic typically runs at midnight but also runs when the Service View service is restarted.

Tiered Service Option Group

A tier is a single row in a tiered service option group. You can use a tiered service option group to produce a variable cost value based on a lookup value. The lookup value from the referencing service option group is matched with the tier values in the tiered service option group.

An association into a tiered service option group starts at the first row and works its way down, with the first tier having the lowest tier values. When an applicable tier is identified and the Quantity Specification is set to System Specified, the lookup value is multiplied by the rates provided in other service option elements in the tier row to determine the amount invoiced. If the Quantity Specification is not set to System Specified, other rates apply.

When a service option element is associated with a tiered service option group, the tier type value determines how the tiered service option group is used. The following tier types are available, depending on the referencing service option element:

Lookup

Use the first tier that matches the value being passed to the tiered service option group.

Lookup Multiple

Use each tier that matches the value being passed to the tiered service option group.

Variable Lookup

Use each tier up to and including the tier that exactly matches the value being passed to the tiered service option group.

Fixed

Use the first tier that matches the value being passed to the tiered service option group. Once in that tier the tier is fixed and cannot change.

Fixed Incremental

Use the first tier that matches the value being passed to the tiered service option group. Increment to the next tier only if the value passed in exceeds the previously used tier. That is, a certain tier level is used; a lower tier level can never be used.

Variable Fixed

This is a merger between variable lookup and fixed tier types. Use each tier up to and including the tier that exactly matches the value being passed to the tiered service option group. Once in the tier that exactly matches, that tier is fixed and cannot change.

Variable Fixed Incremental

This is a merger between variable lookup and fixed incremental. Use each tier up to and including the tier that exactly matches the value being passed to the tiered service option group. Increment to the next tier only if the value passed in exceeds the previously used tier.

Edit a Service Option Group Definition

You may need to update the service option elements in a service option group to meet your needs.

To edit the service option group definition

1. Click the service option group name link.

    The Service Option Group Definition window appears.

Delete a Service Option Group

You can delete an existing service option group.

Important! Deletion is permanent. Deleted service option groups *cannot* be recovered.

To delete a service option group

1. Click Service Builder, Service Option Groups.

    The Service Option Groups page appears.

2. Click the service option group that you want to delete.

    The service option group is selected.

3. Do one of the following *if* the service option group that you want to delete contains inherited service option groups. Otherwise, skip to the next step.

    ■   Delete the child, inherited service option group

    ■   Break the inheritance by editing (see page 174) the child, inherited service option group

    Note: You cannot delete a folder or service that contains inherited service option groups.

4. Click the Delete icon in the Actions column for the selected service option group. Confirm the deletion when prompted.

    The service option group is deleted.

You have deleted the service option group.

Effects of Deleting a Service Option Group

Deleting a service option group (see page 174) affects the subscriptions and requests for the deleted services, as follows:

Subscriptions

The account subscribed to the service option group being deleted no longer contains or lists that service option group.

Requests

The status of requested service options in the service option group being deleted change based on their original status, as follows:

Original Status	New Status
Not Submitted	The service option group and its service options are deleted from the request
Submitted, an approval status, a fulfillment status, Pending Resource Assignment, or Resource Assigned	The service option group and its service options are set to Canceled
Completed	The default cancellation status (Pending Cancellation or Cancel) if CA Service Accounting is installed; Canceled if CA Service Accounting is *not* installed
Pending Cancellation or Canceled	same as original status

Inheritance of Service Option Groups

You can use inheritance to create service option group (see page 176). The inherited (dependent) service option group has the same characteristics as the parent service option group from which it was created. However, the items of the dependent service option elements appear in blue. The blue color indicates that these items are inherited and have not been changed since they were copied.

The following rules apply to the inheritance relationship:

■   If you update the details of the parent service option group, the Catalog system automatically makes the same updates to the dependent service option group. The updated values of the parent replace any existing values in the dependent service option group, except the name.

■   You can manually update important details of the dependent service option group, including its name, definition, and service option elements. However, if you update these specifications in the dependent service option group, you break the inheritance relationship. If this relationship breaks, then changes to the parent *no longer* propagate automatically to the formerly dependent service option group.

When you manually update an inherited item, its color temporarily changes from blue to red. The red color indicates the following:

–   The item was originally inherited but has been changed manually.

–   Consequently, its inheritance relationship is broken.

After the initial manual update, the temporary red color changes to black. The black color indicates that the item is a base (non-inherited) item.

Create an Inherited Service Option Group

You can use inheritance (see page 176) to create a service option group. The inherited service option group is a copy of the original one and is dependent on it.

Follow these steps:

1. Click Service Builder, Service Option Groups.

   The Service Option Groups page appears.

2. Select the service option group that you want to *be inherited by* another service option group and click Copy.

   The service option group is copied so that it can be inherited.

3. Click Paste as Inherited on the header bar of the catalog where you want to place the inherited service option group.

4. Enter the name of the new, inherited service option group.

You have used inheritance to add a service option group.

View Service Option Group Dependencies

To help determine the impact of any changes to a service option group, you can check its dependencies. A service option group dependency displays the following:

■ The services that use a service option group

■ Any accounts that have subscribed to or requested a service that uses a service option group

To view service option group dependencies

1. Select Service Builder, Service Option Groups.

   The Service Option Groups page appears.

2. Click the option button next to the service option group whose dependencies you want to view.

   The service option group is selected.

3. Click the View Dependencies link at the top of the business unit service option group list.

   The Service Option Group Dependency List dialog appears.

4. Select the dependencies you want to view in the List Dependencies By list.

5. Click Search.

   The Catalog system displays the dependencies for the option that you selected.

You have viewed service option group dependencies.

Published Reports

You can publish a report layout (see page 178) that appears as a service option in a special service option group. This service option group is created the first time a report is published. This service option group is named "Published Reports." You can manage it as you manage other service option groups.

The service option representing the published report contains the following:

■ Two Text type service option elements that display the report name and comment values

■ A Rate Type service option element that holds the report cost

You can edit this service option like other service options and include it in catalog services, as required.

Publish a Report Layout

When you publish a report data layout, it appears (as a service option) in a special service option group (see page 177) that is created the first time a report is published.

To publish a report layout

1. Select Administration, Report Builder, Layouts.

   The Report Layout Objects page appears.

2. Expand the folders to find and select the report you want.

3. Click the Publish Layout to the Catalog icon in the Actions column.

   A new service option for the report layout is added to the Published Reports service option group. The new service option has a default cost of 0.

Note: You can remove a published report from the catalog. To do so, de-select the service option from a service that includes it, or remove the service option from the service option group.

Define a Service Option Group

You define a service option group to configure it to meet your requirements.

To define a service option group

1. Click Service Builder, Service Option Groups.

   The Service Option Groups page appears.

2. Click the service option group that you want to define.

3. The Service Option Group Definition dialog appears.

4. Do any or all of the following that apply:

   ■ Add, modify, or delete individual service option elements (see page 179) (cells in the service option group grid).

   ■ Add or modify header text.

   ■ Insert, delete, copy, or paste entire rows and columns of cells.

5. Click Save.

   The Catalog system saves your changes.

You have defined the service option group.

Service Option Elements

Service option elements are the building blocks that you use to define a service option group (see page 178). A service option element is an individual cell within a service option group. A service option group must contain at least one service option element, and can contain many, if necessary. In general, service option elements can have the following characteristics:

Static

Provides fixed information about the service option.

Actionable

Specifies that a request, service, or service option follows a specific approval process, fulfillment process, or both.

Financial

Specifies fixed, tiered or usage-based rates and can be tied to Budget and Planning in CA Service Accounting.

Define a Service Option Element

To define a service option element

1. Navigate to Service Builder, Service Option Groups from the main menu.

2. Click the service option group name to which you wish to add the new service option element.

   The Service Option Group Definition window appears.

3. Double-click the new row or column heading to add a new row or column of cells.

   a. Double-click a new, empty cell to define a new service option element.

   b. Double-click an existing service option element to edit the definition of an existing service option element.

   The Service Option Element Definition dialog appears. You use it to specify the characteristics of a service option element using two tabs, Definition and Options.

4. Complete the service option element fields and click Update Item.

   The service option element is updated with your changes.

Service Option Element Definition

The Service Option Element Definition window lets you specify the characteristics of the service option element. Two tabs contain the fields that define the service option element: the Definition tab and the Options tab.

Service Option Element Definition Window--Definition Tab

These fields on the Definition tab of the Service Option Element Definition window may require explanation.

Item Type

Defines the type of service option element. Available service option element types follow.

Note: For cost and charge-related Service Option Element types, CA Service Accounting is required for invoicing and chargeback.

Adjustment

Adjusts the charge of an associated service option element based on the value indicated. Can be a fixed value or a specified multiplier of the value of the associated service option element.

Application

Specifies a chargeable item based on usage of an associated application.

Boolean

Specifies a true or false value.

CA Business Service Insight Contract

Associates a CA Business Service Insight service level agreement (SLA).

Note: This option applies only if CA Business Service Insight is installed and integrated with CA Service Catalog. For details about this integration, see the *Integration Guide*.

CA Workflow Form or (Form Designer) Form

Associates a form with a service option element which displays when a service option is requested by a user.

Select Form to associate a form created with the Form Designer, or select CA Workflow Form to associate a form created with CA Workflow. Both types of forms are used to provide or gather additional information to complete the request. When you use a form in a service, these rules apply:

- The first service option element in a service option row must not be a Form type.

- The first column of the service option group definition must be a Text type. This setting enables the View Form link to display on the Cart and Request Details page. In a multilingual environment, this setting enables that link to display in all languages.

You can create, customize, and use forms (see page 278) with the Form Designer (the preferred method).

Note: For instructions to create and use forms with CA Workflow, see the *Integration Guide*.

Date

Date values used with tiered service option groups.

Day Of

Similar to a Rate Service Option Element type but also allows a day of the week for weekly charges or a day of the month for monthly charges.

Numeric

Specifies either a fixed numeric value or a numeric value entered by the user requesting the service that contains this service option element

Rate

Specifies a chargeable item not based on application usage.

Reservation

Associates a Reservation Manager reservation of a physical or virtual resource.

Note: This option applies *only* if Reservation Manager is installed and integrated with CA Service Catalog. For details about this integration, see the *Integration Guide*.

Text

Specifies the text that appears to users when they select the associated service option.

This option is required for the following:

■ Adjustment, Text and Form service option element types

■ Agreement type service option elements when the Show Details option is checked

Additional Text Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a Text type service option element.

Text Value

Additional text associated with the service option element.

Associate Service Option Group

Indicates that a service option group is associated with this service option element. Selecting this field displays a list of tiered service option groups, allowing the Administrator to select an associated service option group and which Tier Type is to be used.

Rich Display Text

Converts the Display Text field to a rich text field allowing the Administrator to enhance the appearance of the Display Text field using images and special formatting.

The recommended size for a Service Option Element image is no larger than 48 x 48 pixels.

Important! The following conditions are very important!

Do *not* select Rich Display Text when the Track as an Asset option is selected on the Options tab of the Service Option Element Definition window.

Do *not* select Rich Display Text for the first Service Option Element in a Service Option when using the request system catalog, as this value is used in assigning tasks and email text when using workflow-driven approval and fulfillment processes.

Information Row Only

Indicates that this service option is for information only and cannot be subscribed to or requested. This feature lets you provide information in the catalog, such as additional text or a hyperlink.

Additional Numeric Range Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a Numeric Range type service option element.

Lower Bound

Specifies the numeric value of the lower bound for matches when this service option group is referenced.

Upper Bound

Specifies the numeric value of the upper bound for matches when this service option group is referenced.

Infinite Upper Bound

Specifies that the Catalog system uses this service option element for matches when the lookup value exceeds the value specified for Lower Bound.

Specify Median

Exposes the following field:

Median Value

Specifies that when determining the tier to be used, the value to be compared is the lookup value provided from the usage data. However the value used as a multiplier against any Rate service offering elements for the tier is the absolute value of the difference between the lookup value and the median value.

Additional Rate Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a Rate type service option element. Many of these values apply only if CA Service Accounting is installed.

Cost Type

Specifies the type of cost for the Service Option Element. Select from the following list:

Specify Value

The Administrator specifies the cost value that appears in the catalog and the requesting or subscribing user cannot change the value.

This setting displays the following field:

- Unit Cost: The cost value that is to appear in the catalog.

User Specified

The Administrator specifies the default cost value that appears in the catalog. The requesting or subscribing user can change the value.

This setting displays the following fields:

- Default Unit Cost: The default cost value that is to appear in the catalog

Allocate Cost

The cost is determined by using values associated to a Set as established in Accounting, Budgeting and Planning worksheets. This setting is only applicable if you have CA Service Accounting installed.

This setting displays the following fields:

- Default Unit Cost: This value in this field is set to 0, as the cost is determined from the worksheet value specified for the associated set along with the selected Allocation Method.

- Set: List of Accounting Budgeting and Planning sets available for this cost type. The value of None also appears in the list.

- Allocation Method: The list of possible methods of allocating the cost tied to the service option element.

- Assign: Use the value in the set for the total cost of this service option element for every subscription or request.

- Distribute by Subscribed Account: Use the value in the set for this service option element divided by the number of accounts subscribed to this service option element.

- Distribute by Subscription: Use the value in the set for this service option element divided by the number of subscriptions to this Service Option Element.

- Weighted Distribution: Use the value in the set for this service option element allocated according to actual usage by the account.

Standard Cost

The unit cost is determined by pulling values associated to a Set value as established in Budgeting and Planning worksheets in CA Service Accounting. This setting applies only if CA Service Accounting is installed.

This setting displays the following fields:

- Default Unit Cost: This value in this field is set to 0, as the cost is determined from the worksheet value specified for the associated set along with the selected Allocation Method.

- Set: List of Accounting Budgeting and Planning sets available for this cost type. The value of None also appears in the list.

- Allocation Method: The list of possible methods of allocating the unit cost tied to the service option element.

– Assign: Use the value in the set for the unit cost of this service option element.

– Display Unit Type

Text value that appears with the cost value.

Charge Type

Indicates whether cost value must appear as a Charge or a Credit on an Accounting invoice.

Budget

Indicates whether the service option element appears in the Budget and Planning worksheet.

If CA Service Accounting is not installed, this field serves as additional categorization for the service option element.

Billing Cycle

Indicates how the cost value is applied to an invoice if CA Service Accounting is installed. Select from the following list:

■ One-Time - The charge is applied one time.

■ Installments - Cost is applied on an installment plan.

This setting displays the following fields:

■ Periodic Type: The type of interval to be used when applying the cost: Daily, Weekly, Monthly or N/A.

■ Periodic Type Interval: The frequency of the interval specified in Periodic Type field used to determine the billing interval of the cost.

■ Number of Installments: The number of times the cost must be applied before no longer applying the cost.

Periodic

This setting displays the following fields:

■ Periodic Type: The type of interval to be used when applying the cost: Daily, Weekly, Monthly or N/A.

■ Periodic Type Interval: The frequency of the interval specified in Periodic Type field used to determine the billing interval of the cost.

N/A

■ Not applicable.

Quantity Specification

Indicates how a quantity value must be applied. Select from the following list:

Flat Rate

The cost is applied as a flat rate without alteration.

This setting displays the following fields:

■ Default Quantity Value: The multiplier for the cost value.

Specify Quantity

■ The cost is multiplied by the value specified in the Quantity field.

This setting displays the following fields:

■ Quantity: The multiplier for the cost value.

■ Show Quantity: Indicates whether the Quantity value must be shown in the catalog.

Lookup Admin Specified Quantity

The cost is applied according to an associated service option group.

This setting displays the following fields:

- Service Option Group: Displays the associated service option group chosen from the list of tiered Service Option Groups in the list or None.

- Service Option Element: List of service option elements in the tiered service option group which must be associated with this service option element.

- Show Quantity: Indicates whether the Quantity value must be shown in the catalog.

Lookup User Specified Quantity

The cost is applied according to a specific service's service option element.

This setting displays the following fields:

- Service: List of services or None.

- Service Option Group: List of service option groups for the selected service in the list or None.

- Service Option Element: List of service option elements in the selected service option group which must be associated with this service option element.

- Show Quantity: Indicates whether the Quantity value must be shown in the catalog.

User Specified

The cost is multiplied by the value specified in the quantity field.

This setting displays the following fields:

- Default Quantity Value: The multiplier for the cost value. The user can set this value.

- Show Quantity: Indicates whether the Quantity value must be shown in the catalog.

System Specified

The cost is applied by the catalog system based on a usage quantity.

Note: System Specified only applies when referencing a tiered service option group.

Additional Application Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for an Application type service option element. Many of these values only have meaning if CA Service Accounting is installed.

Pricing Structure

Specifies the method of charging for a selected application. Select from the following list:

Subscription Based

The cost is based strictly on a pre-defined fixed rate. This setting structure displays the following fields:

- Cost Type: The cost type that must be used to apply the cost of this service option element. This field has the same effect as the Cost Type field on the Rate type service option element. That is, depending on the Cost Type selected, additional fields are available.

- Display Unit Type: Text value that appears with the cost value

- Charge Type: Indicates whether cost value must appear as a Charge or a Credit on an Accounting invoice.

Tier Based

The cost is derived from an associated tiered service option group based on a lookup value. This setting displays the following fields:

■ Service Option Group: The list of tiered service option groups, allowing the Administrator to select an associated service option group.

■ Tier Type: The list of available tier types which indicate how the tiered service option group selected is to be used.

Usage Based

The cost is based on the usage information gathered by data mediation.

Note: In order for CA Service Accounting to properly assign costs, the fiscal period of the Budgeting and Planning Set used and the account Billing Cycle must be aligned. For example, if monthly fiscal periods are defined in the set used, the billing cycles of associated accounts must also be set to monthly. In addition, the Period Start and Period End dates for an account's Accounting Profile must be aligned with the fiscal period start and end dates with the account period end date one day greater than the fiscal period end date.

This setting displays the following fields:

■ Cost Type: The cost type that must be used to apply the cost of this service option element. This field has the same effect as the Cost Type field on the Rate type service option element. That is, depending on the Cost Type selected, additional fields are available.

■ Display Unit Type: Text value that appears with the cost value

■ Charge Type: Indicates whether cost value must appear as a Charge or a Credit on an Accounting invoice.

Budget

Indicates whether the service option element appears in the Budget and Planning worksheets.

If CA Service Accounting is not installed, this field serves as additional categorization for the service option element.

Show Metric Result

Shows the metric results on an Accounting invoice as well as provides a link to a report representing this data.

Application

The list of available applications appears in the list.

Metric

The list of metrics available for the selected Application is shown.

Additional Agreement Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for an Agreement type service option element. Many of these values have meaning only if CA Service Accounting is installed.

Note This type of Service Option Element can be used only if CA Business Service Insight is installed.

Show Agreement Details

Includes a link to the Service Level Agreement (SLA) details in the Service Option Element.

SLA Package

Includes all available SLA packages.

Charge for SLA Package

Displays additional cost-related fields that are the same for a Rate type service option element.

Associate Service Option Group

Indicates that a service option group is associated with this service option element. Selecting this field displays a list of tiered service option groups. The Administrator selects a service option group and associates a Tier Type.

Note: This field is only available when the SLA Package choice is None.

Budget

Specifies whether the service option element appears in the Budget and Planning worksheets.

If CA Service Accounting is not installed, this field serves as additional categorization for the service option element.

Additional Numeric Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a Numeric type service option element. Many of these values only have meaning if CA Service Accounting is installed.

Numeric Specification

Specifies whether the subscriber or requester can change the numeric value of the service option element. Select from the following list:

■ Specify Value - The Administrator sets the value of the service option element. This setting displays the following field:

■ Numeric Value: The numeric value of the service option element.

■ User Specified - The Administrator sets the default value of the service option element which the subscribing or requesting user can change. This setting displays the following field:

■ Default Value: The default numeric value of the service option element.

Show Numeric Value

Determines whether the value of the service option element is shown to the user.

Associate Service Option Group

Indicates that a service option group is associated with this service option element. Selecting this field displays a list of tiered service option groups. You can select a service option group and a related tier type.

Additional Boolean Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a Boolean type service option element. Many of these values only have meaning if CA Service Accounting is installed.

Boolean Value

Determines the value of the service option element. Select from: False or True.

Associate Service Option Group

Indicates that a service option group is associated with this service option element. Selecting this field displays a list of tiered service option groups, allowing the Administrator to select an associated service option group and which Tier Type is to be used.

Additional Adjustment Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for an Adjustment type service option element. Many of these values can be used only if CA Service Accounting is installed.

Adjustment Value

Displays the numeric value of the adjustment to the Accounting invoice.

Charge Type

Indicates whether cost value must appear as a Charge or a Credit on an Accounting invoice.

Adjustment Type

Indicates how the Adjustment Value is applied to an invoice if CA Service Accounting is installed. Select from the following list:

- Applied Amount - The actual amount of the associated service option element is applied.

- Multiplier - A multiplier of the associated service option element is applied.

Service

Displays the list of services or None.

Service Option Group

Displays the list of service option groups for the selected service in the list or None.

Service Options Element

Indicates the service option element in the selected service option group that is applied as an adjustment on an Accounting invoice.

Additional Date Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a Date type service option element. Many of these values only have meaning if CA Service Accounting is installed.

Date Type

Specifies the type of date for the service option element. Select from the following list:

■ Specify Value - The Administrator specifies the date value that appears in the catalog. This setting displays the following field:

■ Date Value: The date value that is to appear in the Catalog.

■ Subscription Date - The system sets this value to the date of the request or subscription.

■ Invoice Date - The system sets this value to the date of the invoice when CA Service Accounting is installed.

Associate Service Option Group

Associates this service option element with a service option group. Selecting this field displays a list of service option groups and tier types that you can select.

Additional Date Range Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a Date Range type service option element.

Lower Bound

Specifies the numeric value of the lower bound for matches when this service option group is referenced.

Upper Bound

Specifies the numeric value of the upper bound for matches when this service option group is referenced.

Infinite Upper Bound

Specifies that this service option element is used for matches when the lookup value exceeds the value specified for the Lower Bound.

Additional Service Option Element Fields

Additional fields appear on the Service Option Element Definition tab for a Day Of type service option element. These fields apply only if CA Service Accounting is installed. Most of the fields for a Day Of type service option element have the same meaning as the fields on a Rate type service option element. A Day Of type service option element also includes the following field:

Day of Week

Specifes the day of the week on which to bill the item.

Additional CA Workflow Form Type Service Option Element Fields

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

The following additional fields are available on the Service Option Element Definition window Definition tab for a service option element whose type is CA Workflow Form.

Display Text

Specifies the descriptive text for this form. This text appears to administrators browsing the list of available forms in the Service Builder. Use the Choose Existing file magnifying glass to view the list of forms and select one.

Form Definition File

Specifies the name of the XML file that defines the layout of the CA Workflow form. Use the Choose Existing file magnifying glass to view the list of available forms and select the form.

Upload File

Specifies that you want to upload an XML file to the server, to associate the file with the CA Workflow form service option element.

Note: If you have installed Service View on multiple computers, for optimal efficiency, we recommend that you set up a filestore. A filestore is a single location for shared files, including forms. If you have set up a filestore, verify that this form resides there. For information about setting up a filestore, see the *Implementation Guide*.

Additional Form Type Service Option Element Fields

The following additional fields are available on the Service Option Element Definition window Definition tab for a service option element whose type is Form, meaning that the forms are maintained with the CA Service Catalog Form Designer.

Note: For instructions to create and maintain forms with the Form Designer (the preferred method) instead of CA Workflow, see .

Display Text

Specifies the descriptive text to display for this form to administrators browsing the list of forms in the Service Builder. Use the Choose Existing file magnifying glass to view the list of forms and select one.

Form Name

Specifies the name of the Form Designer form that you want to define as a service option element for this service. Use the Choose Existing file magnifying glass to view the list of available forms and select the form to be used.

Note: Unlike CA Workflow forms, forms maintained using the Form Designer do *not* get stored in the filestore, a single location for shared files. We recommend using a filestore if you are using multiple Service View servers. Form Designer forms are automatically available to all *authorized* users specified by the administrator, as explained in the chapter "Using the Form Designer." However, as a best practice, store *other* shared files besides Form Designer forms in the filestore, as explained in the *Implementation Guide*.

Service Option Element Options Window--Options Tab

These fields appear on the Options tab of the Service Option Element Definition window.

Code

A user-specified text value to represent the product code, subscription code, SKU # or any other applicable code.

Category

Specifies the major category of the service option element from a list of categories.

The Category value is significant in determining which rule actions are executed to determine approval and fulfillment business processes for a request using a service including this service option element.

In addition, this value is used to determine the asset type for assigning an asset. If Track as an Asset is selected and CA APM is installed, the CA APM Assign Asset window can be used to associate a request item with a software or other type of asset.

Note: This field can be set only for the first service option element in any given row.

Category Class

Specifies a value from the list to help further categorize this service option element.

Note: This field can be set only for the first service option element in any given row.

Category Subclass

Specifies a value from the list to help further categorize this service option element.

Note: This field can be set only for the first service option element in any given row.

Track as an Asset

Identifies whether the service option element is eligible to be associated with an asset.

■ If Track as an Asset is selected and CA APM is installed, the CA APM Assign Asset window can be used to associate a request item with a Software asset or other type of asset.

■ If Track as an Asset is selected and CA APM is not installed, this field has no special meaning and can be used for additional categorization.

Important! The following conditions are very important!

Do *not* select Track as an Asset when the Rich Display Text option is selected on the Service Option Element Definition window Definition tab.

Track as an Asset can be set *only* for the first service option element in any given row.

Keywords

Specifies a list of comma-separated list of keywords that are referenced during a catalog search.

Note: This field can be set only for the first service option element in any given row.

External ID

A user-specified text value to represent the product code, subscription code, SKU # or any other applicable code.

Note: This field can be set only for the first service option element in any given row.

URL Info

Displays a clickable URL with the service option element to the requesting or subscribing user.

Display Type

Specifies how the service option element appears in the request/subscription and invoice (if CA Service Accounting is installed). This setting allows you to hide the service option element from view in special cases. Select from the following list:

■ Include in Request/Subscription and Invoice

■ Include in Request/Subscription, Exclude from Invoice

■ Exclude from Request/Subscription and Invoice

Change to Take Effect

Depending on the Catalog Configuration setting for Default Effect of Service Option Element Changes additional choices may appear as service option elements are added to or changed for a service option group that is already included in a service.

Select from the following list:

■ Beginning of Accounts' Current Billing Period - No Audit Trail - The change takes effect retroactively to the beginning of the current billing period for existing subscribers or requesters.

■ Beginning of Accounts' Current Billing Period - The change takes effect retroactively to the beginning of the current billing period for existing subscribers or requesters.

■ Beginning of Accounts' Next Billing Period - The change takes effect at to the beginning of the next billing period for existing subscribers or requesters.

■ Immediately during Accounts' Billing Period - The change takes effect immediately for existing subscribers or requesters.

■ Specify a Future Effective Date - The change takes effect for existing subscribers or requesters on the date specified.

Design a Form to Associate to a Service Option Element

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

You can use a CA Workflow form to associate a service option element. If you use CA Workflow forms, for best results, store the forms in a separate process definition and use this process definition only to save and access forms. A process definition named USM_Forms_Container is provided for this purpose.

To design a CA Workflow form to associate a service option element

1. Click Administration, Tools, Links, CA Workflow, IDE.

2. In the Process Manager, Process Definitions menu, double-click the process definition name to open it for editing.

   The Process Designer window appears.

3. Click Forms.

   The list of existing forms appears.

4. Click Add.

   The CA Workflow Form Designer starts. You can drag and drop the following elements to create a form:

   Group Box

   Groups other element in a section on the form.

   Input

   Specifies the inputs and apply attributes to the form.

   Secret

   Specifies a password and keeps the value hidden from view.

   Output

   Specifies the display-only fields on the form.

   Text Area

   Lets you have a free-form text area.

   Select

   Can be used as a multiple select list box or as a set of check boxes. A multiple list box displays an expanded select element so that several options appear. You can select them when holding down the Shift key. To turn this multiple list box into check boxes, change the Appearance attribute.

   Note: Each element ID in the form must be unique. For more information on each type of element, see the *CA Workflow IDE Online Help*.

5. Click Preview to preview the design.

6. Export the form to the client computer, when you have finished the design.

   The file is exported as an XML file.

7. Click OK.

   The CA Workflow Form Designer closes.

8. Click Save on the Process Designer window.

   The process definition which includes the form is saved.

Some of the form element attributes having an effect on the form include the following:

Readonly

Makes the text box element read-only.

Required

Requires the requester to complete the form before continuing with the request process.

Hint

Creates a tool tip for the text box when the mouse hovers over the element.

Help

Adds an extra row beneath the element with more information to help the user understand and complete this element of the form.

Type

Specifies whether the service option element is hidden. When hidden, it does not appear on the form.

Value

Omits a default value. The text box can remain empty.

Id

Specifies the unique ID for the form. This parameter does not appear on the form.

Label

Specifies the label to appear next to an element to describe its purpose.

Size

Specifies the length of the text box element.

Cols

Specifies the number of columns (width) for this text area.

Rows

Specifies the number of rows (height) for this text area.

itemValues

Specifies the values of the selected items. This is a multi-valued string separated by commas or semicolons.

itemLabels

Specifies the display values for the items   This is a multi-valued string separated by commas or semicolons.

Appearance

Specifies *one* of the following:

Full: Vertically aligned option buttons

Minimal: Horizontally aligned option buttons

Compact: Single select drop down list.

Forms Variables

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

Variable replacement is supported so that default values can be pre-populated with specific information regarding the user, business unit, service, or service option group. This allows an administrator to set up the form so that the default value can reflect the logged in user's unique information.

The supported variables are composed of two pieces, the object and the object property. The object properties are case sensitive. Reference the variables within the $ sign symbols.

The supported objects are as follows:

user

Specifies the logged in user object.

bu

Specifies the business unit of the role with which the user logged in to CA Service Catalog.

service

Specifies the service object requested.

sog_name of service option group

Specifies a service option group defined in the service object requested.

Examples: sog_Local Printer, sog_Network Printer

The properties are as follows:

user object

An example: "$user.postalCode$"

user_id, first_name, middle_name, last_name, given_name, common_name, alias, title, manager, default_domain (business unit), default_role, address1, address2, address3, address4, address5, address6, mailing_address1, mailing_address2, mailing_address3, mailing_address4, mailing_address5, mailing_address6, city, state, country, county, postalCode, location_name, location_comment, location_uuid, location_phone, location_fax, primary_phone, secondary_phone, telephone, fax, pager, email, mobile, status, comment, time_zone_code(in English only), request_account

bu object

An example: "$bu.tenant_name$"

domain_type (business unit type where SP=service provider, ST=can have sub-business units, TE=cannot have sub-business units), single_account_mode (0=false, 1=true), status, tenant_id (business unit ID), tenant_name (business unit name), opened_date, email, website, description, federal_tax_payer_id, state_tax_payer_id, tax_region, time_zone_code (in English only), date_format, time_format, decimal_format, parent_tenant_id (parent business unit ID), data1, data2, data3, data4, data5, data6, data7, country_name, state_name

service object

An example: "$service.offering_name$"

offering_id, date_available, date_unavailable, date_created, date_cancelled, code, offering_name, description, domain

sog object

An example: "$sog_Local Printer.description$"

rate_plan_id, date_available, date_unavailable, date_created, date_cancelled, code, rate_plan_name, description, domain

How to Pre-Populate Select Controls in Forms

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

To pre-populate the fields in a form created in CA Workflow, perform the following tasks:

1. In Report Builder, create and verify a query that addresses your need or goal.

   If necessary, create the data objects (see page 99) and runtime variables needed for your query.

   When you use them in forms, you specify methods for pre-populating select controls (see page 204).

2. In CA Workflow, do the following:

   a. Create the form that uses your new query. You create a form when you design a form to associate to a service option element (see page 199).

   b. Associate the report data object to the form element.

      Note: For details, see the CA Workflow IDE Help.

3. In CA Workflow, do one of the following:

   ■ Export the form to %USM_HOME%/view/forms.

     If you export the form to this location, you do not need to upload it from Service Builder.

   ■ Export the new form to a different location that you can access from Service Builder.

   Note: For details, see the CA Workflow IDE Help.

4. In Service Builder, do the following:

   a. If you exported the form to %USM_HOME%/view/forms, skip this step and go to the next step (b).

      If you exported the form to a different location, upload the form now. You upload the form when you design a form to associate to a service option element (see page 199).

   b. Add a new service (see page 142) and add a new service option group (see page 171) in which to include the form.

      Alternatively, optionally use an existing service (see page 141) and service option group (see page 169), if they meet your criteria.

   c. Attach the form to the service option element of your choice. You attach the form when you define a service option element (see page 179).

5. As a catalog user, shop from the catalog (see page 564), complete a request for the service, and verify that the form works as planned.

6. If necessary, refine the query, the form, or both until they work as planned.

   For details, see the sections previously mentioned in this overview.

The scenario for using lookup parameters to pre-populate form fields (see page 208) is an extended example that illustrates both the main tasks in this process and the lookup parameters.

Methods for Pre-Populating Select Controls

Select controls are used to present option buttons and drop down lists on a form. It is possible to include either static or dynamic entries in the select control of a form. When users request a service and enter data in a form for the request, these select controls appear as drop-down lists, list boxes, or option buttons. As an administrator, you have these options for pre-populating the lists created by a select control in a form. When user enters data in the form, the select control determines the list of options from which the user can choose. The following types of select controls exist:

■ Static entries

■ Variable entries generated by a report data object

■ Variable entries generated by a report data object, using lookup parameters

Static Entries

Use the itemLabels and itemValues fields to populate lists of static entries. These entries can include the "user" and "bu" object variables available in context.

For details, see the Forms Variables sub-section in Design a Form to Associate to a Service Option Element in the chapter "Using Service Builder."

Variable Entries Generated by a Report Data Object

You can also include a variable number of entries that are the result of the output from a report Data Object. To do this, you set the select control's value parameter to "$report_data.*dataobjectID*$" where *dataobjectID* is the report Data Object's object ID. You can obtain the Data Object's object ID from the Edit Data Object window.

The first column returned from the Data Object is added to the itemValues list and the second is added to the itemLabels list. If only one column is returned, it is used for both the itemValue and itemLabel.

A select control resulting in multiple returned values is more likely and is typically more useful. For example, suppose you want a select control named "cities" on a form to hold a list of cities. In that case, consider including an itemLabels value of "-Select City-" to provide one static entry directing the user to select a choice from the list.

Next, consider creating a Data Object named "Available Cities" that returnd the list of cities. Set the value parameter of the "cities" select control to $report_data.0c6f2fc58588f5f4-87e704-10d8e8cfd5f-7ffd$, using the object ID for the "Available Cities" report data object.

Thus, the report data object is static but the SQL query is not. Each time you run the static report data object, it generates a unique SQL query.

When a user begins to specify data on the form, the resulting select control displays the heading "-Select City-" followed by the list of cities provided by the data object.

Variable Entries Generated by a Report Data Object, Using Lookup Parameters

Lookup parameters are also named on-demand input variables. You can use lookup parameters as part of the report data object that pre-populates a select control. With lookup parameters, you prompt the user to supply data that is used as selection criteria in the SQL query that populates the report data object. Consider the example in the previous section. In that example, you use the same exact report data object in every form.

In contrast, when you specify lookup parameters, the report data object is a *variable* whose actual value is generated dynamically at runtime based on user input. One application is an extension of the example in the previous section: You can use lookup parameters to prompt the user for the country and state, which are used to dynamically create a report variable that populates the select control with *only* the names of the cities in that state. Consequently, the Cities drop-down list presented to the user is much more meaningful than a single, potentially very long list of all cities in the country. Moreover, such a custom list may increase system performance significantly in an environment with several hundred or several thousand users spread across multiple geographic regions.

When Catalog users specify data on a form field created by a report data object using lookup parameters, the users see the field name next to an empty drop-down list linked to a search icon (typically a magnifying glass). When users click the search icon, they are prompted to enter the lookup parameters. For example, you may include an Approver field with a lookup parameter that prompts the user to specify the department of the approver. Based on the department that the user selects, the drop-down list next to the approver field is populated with the names of members in that department. Thus, for the approval field on a travel request form, you may use lookup parameters to prompt the user to select the department that must approve the request, for example, the department to be billed for the travel.

Note: For an extended example that illustrates how to use lookup parameters, see Scenario for Using Lookup Parameters to Pre-populate Form Fields in the chapter "Using Service Builder."

Upload a Form

Forms can be uploaded using the Service Option Element Definition window or by manually copying the file to the server.

To upload forms using the Service Option Element Definition window

1. Select the Upload Form check box.

    The upload feature appears.

2. Click Browse to look for a form file to be uploaded from your computer.

3. Click Upload.

    The form uploads.

4. Click Choose Existing File icon associated with Form Definition File field.

    The list of definition form files are uploaded to the server.

To upload forms manually

1. Place the form XML files in the following directory:

    %USM_HOME%\view\forms

2. Create the folder if one does not exist.

3. Copy the XML form files to this location.

    The form is now available in the Service Option Element Definition window for use when defining a service option group.

Note: You can find sample forms in %USM_HOME%\catalog\examples\forms.

Update a Form

To make changes to a form, you can either replace the form with an updated version or upload a new form. For small changes, such as adding a new field, repositioning existing fields, or changing labels, you can replace the file directly on the server. However, doing so may affect any requests that are in progress.

If the changes are more significant, then we recommend that you upload a new form and manually edit all the Service Option Elements that reference this form. Doing so affects future requests.

Delete a Form

To delete a form

1. Open Windows Explorer on the Service View computer that stores the forms.

2. Expand the folder to %USM_HOME%/view/forms.

3. Select the XML files and press the Delete key.

   The selected forms are deleted.

Scenario to Pre-populate Fields

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

This scenario illustrates how to use lookup parameters to pre-populate fields in CA Workflow forms. This scenario also illiustrates the process to pre-populate select controls (see page 203). This scenario shows how you can use the selection criteria for the country in your initial user prompts. Using the country, you can specify lookup parameters so that the resulting SQL query pre-populates the list of locations in the form field with *only* the locations applicable to the user's country. Consequently, you maximize performance.

1. Define all business units for your organization, including all locations throughout the world.

2. Create a data object that prompts users for their country.

3. Select a form and modify it to include your new data object.

4. Specify a service and service option group in which to include your form.

5. Create and test a request that includes your form.

Important! Repeat these steps for each form that you need to create.

Define Business Units and Sub-Business Units Worldwide

Before you start using lookup parameters to pre-populate form fields, verify that you have defined *all* business units and sub-business units in your organization throughout the world. Meeting this requirement is critical to obtaining accurate and complete results when you use lookup parameters.

In CA Service Catalog, you specify and organize locations *only* through your definition of business units and sub-business units. A business unit or sub-business unit may have one or more locations, depending on the structure of your organization and your implementation of CA Service Catalog.

Important! When specifying locations in any context for CA Service Catalog, ensure that you specify the country name carefully and consistently. For example, if your database is configured to be case-sensitive, ensure that you specify the country name of the United States as *United States*, not *united states*, *UNITED STATES*, or some other variation. Meeting this requirement is *especially* important when you define the location or locations of a business units and sub-business units.

Note: For details about defining locations as you create and maintain business units and sub-business units, see Add a Business Unit in the chapter "Managing Business Units and Accounts."

Create a Data Object

Creating a data object that prompts users for selection criteria is the central task required to efficiently use the feature of dynamically pre-populating form fields. In this scenario, your selection criterion is the name of the country, so you prompt users to enter the name of their country.

To create the data object that prompts users for their country

1. If necessary, log in to CA Service Catalog as an administrator.

2. From the home page, select Administration, Report Builder, Data Objects.

3. Click the Create Data Object button.

    The Create Data Object dialog appears.

4. In the Create Data Object dialog, specify the following:

    a. For Type, select Query.

    b. For Database Name, enter mdb.

    c. For Table, enter ca_location.

    d. For Fields, enter the following:

    $location_uuid$, $location_name$

    $location_uuid$ is a column in the ca_location table.

    $location_name$ is also a column in the ca_location table. This name is the location returned when the SQL is executed.

    e. In to the Query box, enter the following:

    SELECT * from ca_location l, ca_country c WHERE l.country=c.id and c.name='%COUNTRY%'

    Note: '%COUNTRY%' is the system variable for country ID.

    This query checks the tables for the location information.

5. Click the Create Variable button.

    The Save Report Variable/Edit Runtime Variable dialog appears.

6. In this dialog, specify the following:

   a. Input for Name: COUNTRY

   b. Input for Datetype: String

   c. Input for Prompt: Enter Country Name:

   d. Click Create Variable.

      You receive a confirmation message that the variable was successfully created.

      If you receive an error message, ensure that you are entering the name of the country exactly as it is defined in the business unit or sub-business unit to which it belongs.

      Note: For details about creating and maintaining locations in your business units, see Add a Business Unit (see page 50).

   e. Click Save & Test.

   The Save Data Object pop-up appears.

7. In the Save Data Object pop-up, do the following:

   a. In the Enter for Name: field, specify Get Location Name.

      Your entry becomes the name of the new data object.

   b. In the Enter for Comment: field, specify Search location name when country name is given.

   c. Optionally, specify the folder in which to save the new data object, using the folder browse icon.

      Otherwise, the new data object is saved in the Data Objects default folder.

   d. Click Save.

   The Enter Parameters dialog appears and informs you that the new data object requires parameters to be specified.

8. In the Input: for Enter Country Name: field, enter the name of a country that you have defined, and click OK.

   For example, if you have defined the United States, enter United States, and click OK.

   The following columns appear: _index, location_uuid, and location_name.

9. Go back to Data Objects page and expand the Default folder.

10. Click the Edit icon associated with Get Location_Name.

11. Verify that the ID generated for the report object appears similar to the following example:

    514e36423b657f1b6235b113dfd8596a-7fb21184956467359

    Remember that each ID is dynamically generated by the SQL query and therefore is unique.

    You are ready to include your new data object in one or more forms. The following sections illustrate how to do so most efficiently.

Include your New Data Object in a Form

Select a predefined form and modify it to include your new data object. You must use a form to include a data object in a request for a service. This procedure is a custom example of designing a form to associate to a service option element (see page 199).

Note: For more information about designing forms, see the *CA Workflow IDE Online Help*. For more information about associating a report data object to a form element, see the *CA Workflow IDE Online Help*.

To select a form and modify it to include your new data object

1. If necessary, log in to CA Service Catalog as a Service Delivery administrator.

2. On the home page, select Administration, Tools, CA Workflow.

   The CA Workflow IDE appears.

3. Click the Process Manager side tab, and click the Process Definitions icon.

   The Process Definitions appear in the main window, displaying the name, short name, modified date, and so forth for all process definitions.

4. Click the Name column to sort the process definitions alphabetically by name.

5. Find and double-click the process definition named USM_Forms_Container.

   The CA Workflow diagram for this process definition appears in the main window.

   Note: If this container has not been modified since CA Service Catalog was installed, then the container stores only boilerplate elements and therefore is nearly empty.

6. On the bottom portion of the screen, under the CA Workflow diagram, click the Forms tab.

   An empty Forms Designer template appears.

7. Click Add.

   The Forms Designer dialog appears.

8. Click the Import button.

    The Open dialog appears, prompting you to select the location of the form that you want to import.

9. Browse to the folder named %USM_HOME%/view/forms.

10. Click AccessToServer.xml and click Open.

11. When asked whether to replace the existing form, click OK.

    The Form Designer: *form name* dialog appears. In this scenario, the Form Designer: Server.NT.Server.Access dialog appears.

    The lines under the form title contain the existing fields and their select controls, in the order that they appear on the form.

12. On the left tool bar, click and drag the "Select1" icon.

    Select1 is typically the last select control in the list. When you mouse-over the Select1 icon on the left tools, the tool-tip text that displays is "Select1 control is a group of items from which the user selects only one."

13. Drag the Select1 icon to the lines that contain the prompts and controls, and drop the icon between the lines that contain the prompts named "Please select the share name" and "Enter the pmfkey of someone that currently has access:"

    Note: The *pmfkey* is another term for *user ID*. Use the appropriate term for your company.

    A new line named Select1 appears.

    Your screen appears similar to the following:
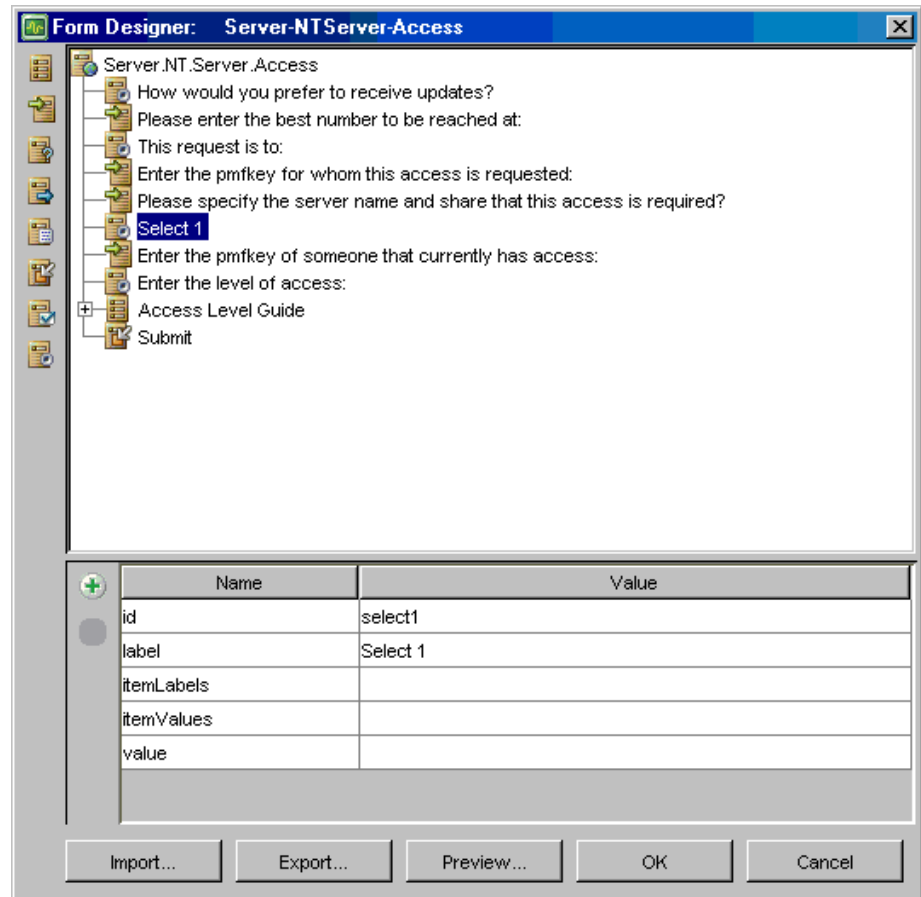
14. Select the new line named Select1.

    The set of names and mostly empty values for Select1 appear, in the table under the form lines.

    Note: On the Form Designer, when you select any line, the names and values appear for the corresponding control in the table under the form lines.

15. In that table, specify the following values for the new Select1 control.

    Double click each field in the Value column to open the field and enter text.

    id

    Enter a unique ID, such as location_name.

    Note: Each element ID in a form must be unique. For more information about IDs, see the CA Workflow IDE Online Help.

    label

    Enter the following:

    Please select the server location:

    This text becomes a new prompt on the form that you are updating.

    itemLabels

    Do *not* enter any text in this field. It is populated dynamically when users complete the form.

    itemValues

    Do *not* enter any text in this field. It is populated dynamically when users complete the form.

    values

    Enter $report_data.*ID*$, where *ID* is the report data object ID generated in the previous section. An example follows:

    $report_data.514e36423b657f1b6235b113dfd8596a-7fb21184956467359$

16. Click Preview.

    The preview shows how your form appears to users.

17. Make any necessary adjustments. When you are finished, Click Export.

    The browse dialog appears.

18. Browse to the %USM_HOME%/view/forms folder and save the form with a new file name, such as AccessToServer_2.xml.

    Important! Use a different file name, to prevent your changes from being accidentally overwritten by a default form included in a later patch or upgrade. Keeping the original form intact is also helpful in case you need to verify the default contents of the form later for any reason.

19. Click OK on the Form Designer.

20. Check the Form Designer and verify your change on the form, which was to add one parameter that says "Please select the server location:" and specify details of its select control.

You are ready to include this form in a service or service option group, so that you can test it in an actual Catalog request.

Add Your Form to a Service Option Group and a Service

Each form is typically used in a service, and each service requires a service option group. Therefore, you must specify a service and service option group in which to include your new form that contains the field for dynamically populating a select control.

To include your form in a service option group and a service

1. Log in to CA Service Catalog as a Service Delivery Administrator.

2. On the home page, click Service Builder, Services.

3. Click Add to create a service.

   The Folder Details dialog for the new service appears.

4. Click the Service option button, and enter a Name for the service, such as Test Dynamically Populated Form Variable.

5. In the Approval Process drop-down list, select Workflow driven approval process.

6. Click OK to save your changes and close the dialog.

   The new service appears in the list of services on the Service Builder, Services page.

7. Click the Define icon for the service.

   The list of all services is replaced with the list of all service option groups.

8. Click Add New Service Option Group.

   The Details dialog for the new service option group appears.

9. Specify the following:

   a. For Name, specify the name that you entered for the service in Step 4.

   b. For Type, select Fixed.

   c. For Selection Type, select Choose Many.

10. Click OK to save your changes and close the dialog.

    The new service option group that you created appears in the list of all service option groups.

11. Scroll down to the new service option group and click it to open it.

    The Service Option Group Definition dialog appears. It displays the rows, columns, and cells (all empty by default) that define the service option group.

12. In Row 1, Column 1, double-click the text labeled Dbl Click To Add Value.

    The Service Option *Element* Definition dialog appears.

13. On the Definition tab, in the Type drop-down list, select Text.

    The dialog expands to include additional fields related to your selection.

14. Supply values in to these fields to meet your requirements.

    For details about these fields, see Service Option Element in the chapter "Using Service Builder."

15. Click the Options tab.

    The fields for this tab appear.

16. Supply the values that you require for Category, Category Class, and Category Subclass.

    For details about these fields, see Service Option Element in the chapter "Using Service Builder."

17. Click Update.

    The Service Option Element Definition dialog closes, and you return to the Service Option Group Definition dialog.

18. In Row 1, Column 2, double-click the text labeled Dbl Click To Add Value.

    The Service Option Element Definition dialog appears.

19. On the Definition tab, in the Type drop-down list, select Form.

    The fields for this tab appear.

20. For Display Text, enter the name you want. Optionally enter same name that you specified for the service option group in Step 9.

21. Click the Choose Existing File icon (typically a magnifying glass icon) to select the form definition file to associate with this service option element.

    A new dialog appears and lists the available forms.

22. Click the form name you want, for example, AccessToServer_2.xml.

    You return to the Service Option Element Definition dialog.

23. Click Update.

    You return to the Service Option Group Definition dialog.

24. Click Save.

    The Service Option Group Definition dialog closes, and you return to the list of all service option groups.

25. Scroll down to your new service option group.

26. Check the Include box next to your new service option group.

    This action includes it with the new service that you created earlier in these steps.

    A new line appears under your new service option group.

27. On that new line, check the Include and Default columns.

28. Scroll to the top of the list of service option groups and click Save Selection.

    A confirmation message appears.

29. Click OK to confirm.

    You return to the list of all services on the Service Builder, Services page.

    You are now ready to create and test a request that uses the service and service option group associated with the form.

Test a Request

To ensure that the dynamically populated variable works as required in your new form, create and test a request that uses the form.

Note: For details about the information in this section, see <u>Shop from the Catalog</u> (see page 564).

To create and test a request that includes your form

1. Log in to CA Service Catalog as a Catalog user.

2. On the home page, select Requests.

    The Requests page appears.

3. In the Browse pane, click Not Categorized.

    The Not Categorized page appears.

4. Click the service that you created earlier, such as Test Dynamically Populated Form Variable.

    The service opens and displays its service option groups.

5. Check the box next the name of the service option group (such as Test Dynamically Populated Form Variable) and click Next.

    The form for the service option group appears, with empty fields to be completed by you. The parameter that contains the form variable must have a search icon (typically a magnifying glass) next to it.

6. Click the Search Icon next to that parameter. In this case, the parameter is "Please select the server location"

    The search dialog appears; in this case, the Enter Parameters dialog appears.

7. In the Enter Country Name field, enter the name of a country that you have defined, and click OK.

    For example, if you have defined the United States, India, or Lebanon, enter that name, and click OK.

    The "Please select the server location" drop down is populated with all the location names defined for the country you specified.

8. Select the name of the location that you want.

9. Specify values for all remaining fields, drop downs, or option buttons on the form.

    As you complete the form, do the following:

    a. Verify that each dynamically populated form variable includes a search icon.

    b. Verify that when you click a search icon, the Enter Parameters dialog for the adjacent variable appears and displays the search values that you have specified when you created the variable.

    c. Verify that the tool tip text for each search icon is "Lookup Parameters" or whatever text you specified.

10. Double-check your specifications on the form.

11. Click *either* Add to Cart *or* Add to Cart and Checkout.

12. When your shopping cart appears, do the following:

    a. Locate the service that you requested and click View.

    b. Verify that the form is editable.

13. Correct any errors that you find and test again until the service and the form yield no errors.

14. Notify the appropriate users and tell them the location of the service and the form.

Having multiple users test the service and form helps ensure that the service and form are ready for all applicable users in your environment.

How to Pre-Populate Text Fields in CA Workflow Forms

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

To pre-populate the text fields in a form created in CA Workflow, perform the following tasks:

1. Design the form conceptually, from start to finish.

    This process includes deciding which fields to pre-populate, deciding which field to use as the key "input" field, and planning a report data object to fetch the required data. The key input field is used for retrieving the data to insert in the fields that you want to pre-populate.

    For example, in an existing form, you may decide to use the user ID field as the key input field. Based on the user ID that the user enters, you create a report data object to retrieve personnel data for that user ID, such as email address, physical address, city, state, phone number, and so forth. You use that data to pre-populate the corresponding fields on the form. Consequently, the user does not need to enter any of that data manually, which improves the accuracy and efficiency of the form.

    Similarly, you may decide to use other important fields to pre-populate additional text fields. For example, if your database stores physical inventory for your organization's computers, you may design an asset transfer form that uses the computer name field to pre-populate data for the RAM, hard drive, bus, peripheral devices, and other data for that computer. The possibilities are numerous and vary widely, depending on your organization's unique resources and needs.

2. In Report Builder, create and verify the report data object.

    If necessary, create the data objects and runtime variables needed for your query.

    Using the first example described in the previous step, you may decide to create the query, using a SELECT-JOIN statement to locate the user ID entered by the user and retrieve the corresponding data for email address, physical address, and so forth. The name of each database field in the query must *exactly* match (including case) the id of the form field that it pre-populates. In this example, for the key input field on the form (user ID), both the id on the form and field name in the database must have the same value (userid).

    For details about data objects and runtime variables, see Data Objects (see page 99).

3. In CA Workflow, do the following:

   a. Create or update the form that uses your new query.

      For details about creating forms, see Design a Form to Associate to a Service Option Element (see page 199).

   b. Associate the report data object to the form element.

      For details about associating report data objects to the form elements, see the CA Workflow IDE Help.

   c. Define all fields on the form, including the pre-populated text fields and the key field used to populate them.

      For details about defining these fields, see Modify a Form to Include New Text Fields (see page 221).

      For details about defining the required attributes for pre-populated text fields, see Define Attributes for Pre-Populated Text Fields (see page 226).

      For more details about creating fields, see the CA Workflow IDE Help.

   Important! The sections mentioned above provide *both* general instructions *and* a running example to explain how you pre-populate fields in forms, using a default form as the starting point. The running example uses the Employee ID (user ID) as the key input field. The running example, named the Employee ID example, is *not* intended as an end product to "plug into" your system as-is but rather as an illustration to demonstrate how you can modify the default forms to match your organization's unique resources and needs.

4. In CA Workflow, do one of the following:

   ■ Export the form to %USM_HOME%/view/forms.

      If you export the form to this location, you do not need to upload it from Service Builder.

   ■ Export the new form to a different location that you can access from Service Builder.

   For details, see the CA Workflow IDE Help.

5. In Service Builder, do the following:

   a. If you exported the form to %USM_HOME%/view/forms, skip this step and go to the next step (b).

      If you exported the form to a different location, upload the form now.

      For details about uploading the form, see Upload a Form (see page 207).

    b. Create a service and service option group in which to include the form.

       For details, see the sections <u>Services</u> (see page 141) *and Service Option Groups* (see page 169).

       Alternatively, optionally use an existing service and service option group, if they meet your criteria.

    c. Attach the form to the service option element of your choice.

       For details, see <u>Define a Service Option Element</u> (see page 179).

6. As a CA Service Catalog user, complete a request for that service and verify that the form works as planned.

    For details, see <u>Shop from the Catalog</u> (see page 564).

7. If necessary, refine the query, the form, or both until they work as planned.

    For details, see the sections previously mentioned in this overview.

Modify a Form to Include New Text Fields

When needed, you can select a form and modify it to include new text fields. For illustration, this topic uses the form named Employee Termination (EmployeeTermination.xml) and continues the Employee ID example started in <u>How to Pre-Populate Text Fields in Forms</u> (see page 219). Employee Termination is one of many forms included by default when you install CA Workflow from the CA Service Catalog installation media.

To select a form and modify it to include your new data object

1. If you have not already logged in to CA Service Catalog as a Service Delivery administrator, do so now.

2. On the home page, select Administration, Tools, CA Workflow.

    The CA Workflow IDE appears.

3. Click the Process Manager side tab, and click the Process Definitions icon.

    The Process Definitions appear in the main window, displaying the name, short name, modified date, and so forth for all process definitions.

4. Click the Name column to sort the process definitions alphabetically by name.

5. Find and double-click the process definition named USM_Forms_Container.

    The CA Workflow diagram for this process definition appears in the main window.

    Note: If this container has not been modified since CA Service Catalog was installed, then the container stores only boilerplate elements and therefore is nearly empty.

6. On the bottom portion of the screen, under the CA Workflow diagram, click the Forms tab.

   An empty Forms Designer template appears.

7. Click Add.

   The Forms Designer dialog appears.

8. Click the Import button.

   The Open dialog appears, prompting you to select the location of the form that you want to import.

9. Browse to the folder named %USM_HOME%/view/forms.

10. Click EmployeeTermination.xml and click Open.

11. When asked whether to replace the existing form, click OK.

    The Form Designer: *form name* dialog appears. In this scenario, the Form Designer: emp_termination dialog appears.

    The lines under the form title contain the existing fields and their attributes, in the order that they appear on the form. When you select a field, the names and values of its attributes appear at the bottom of the form, as shown in a later step.

12. On the left tool bar, the CA Workflow form controls appear in a vertical list. Locate the "Input" control.

    This control adds a new text field. Input is typically the second select control in the list. When you mouse-over the Input control on the left tools, the tool-tip text that displays is similar to the following: "Input control to display as a field to enter information." The Input control icon appears as follows:

13. Click and drag the Input icon to the lines that contain the existing controls and field names, and drop the icon between the lines that contain the fields named "Employee Name" and "Employee Manager Name"

A new line named Input appears; it is the new text field in this form. Your screen appears similar to the following:

14. Select the new field named Input.

The names and values of the attributes of the new field appear, in the table under the form lines.

Note: On the Form Designer, when you select any line, the names and values appear for the corresponding control in the table under the form lines.

15. In that table, specify the values for the attributes of the new field.

   Double click each field in the Value column to open the field and enter text.

   id

   Specifies a unique ID.

   To continue the Employee ID example started in How to Pre-Populate Text Fields in Forms (see page 219), specify first_name, because this new field is the First Name field.

   Note: Each element ID in a form must be unique. For more information about IDs, see the CA Workflow IDE Online Help.

   label

   Specifies an intuitive label.

   In the Employee ID example, the label is First Name.

   This text becomes the label that users see when they view this field on the form.

   value

   Specifies a default value such as some text or a number, variable, report data object, or JavaScript function, depending on the purpose of this field.

   If none of these options applies to the field you are creating, leave this attribute blank, so that this field is completely empty when users open the form.

   In the Employee ID example, leave this value blank, because it is pre-populated later.

   When you are finished, click OK.

16. Using the previous few steps as a model, use the Input control to create and place any other text fields that you want to add to this form.

For each new field, specify the id, label, and attributes, as described in the previous step.

To continue the Employee ID example, define the following fields on the form and place them sequentially after the First Name field. For the field attributes, use the specifications shown.

- Last Name: id = last_name; label = Last Name

- Phone Number: id = phone_number; label = Phone Number

- Email Address: id = email_address; label = Email Address

- Physical Address: id = physical_address; label = Physical Address

Like the First Name field, these fields are to be pre-populated. Remember that the id of each field must *exactly* match (including case) the name of the corresponding field in the database that you are querying.

Also create the user ID field if you did not already create it in the previous step.

When you are finished, click OK to close the Form Designer dialog and save your changes.

17. To finish the Employee ID example, define the *additional required attributes* for the key field (user ID) used to pre-populate the other fields. For details, see Define Attributes for Pre-Populated Text Fields (see page 226).

Define Attributes for Pre-Populated Text Fields

In CA Workflow, you can create text fields on forms as needed. To pre-populate text fields, you must follow the steps in this section. These steps include adding several new attributes required for the key field, the field used to capture the data used to pre-populate other text fields. This topic continues the Employee ID example started in How to Pre-Populate Text Fields in Forms (see page 219).

To define attributes for pre-populated text fields

1. Ensure that you have selected the form you want to use and that the fields you want to pre-populate are already created. If you need to create any of these fields, do so now, as explained in Modify a Form to Include New Text Fields (see page 221).

2. Ensure that the id of each pre-populated text field matches *exactly* (including case) the corresponding database field in the query for the report data object, as explained in How to Pre-Populate Text Fields in Forms (see page 219).

    To continue the Employee ID example started in that section, ensure that the following text fields are created, and that their id attributes on the form exactly match the corresponding field names in the database: First Name, Last Name, Phone Number, Email Address, and Physical Address.

    If necessary, follow the steps in Modify a Form to Include New Text Fields (see page 221) to create these fields and specify their attributes.

3. Open the form that you want to edit. For details, if necessary, see Modify a Form to Include New Text Fields (see page 221).

4. On the form, select the key input field.

To continue the Employee ID example, open the Employee Termination form and select the Employee ID field.

The existing attributes for that field appear in the field attribute table on the bottom part of the screen: The Name and Value columns display the names and values of each attribute for the selected field. The names of the default attributes for the Employee ID field are id, label, required, and value. Your screen appears similar to the following:
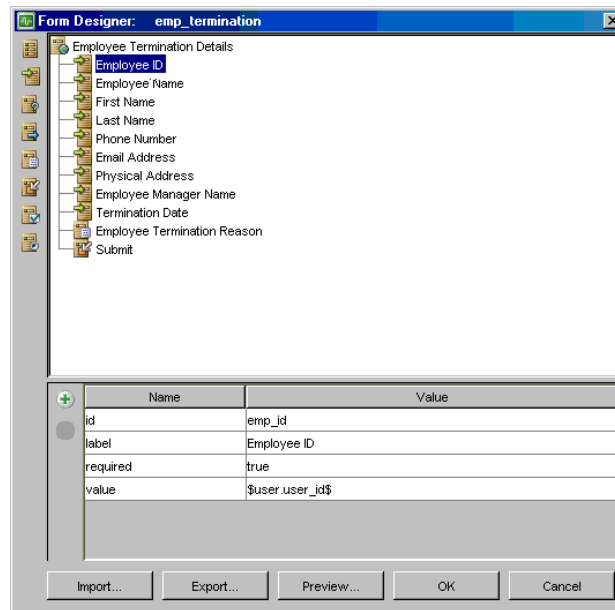
5. To create a new attribute for the selected field (Employee ID), click the green plus (+) sign icon next to the Name column of the field attribute table.

6. In the row for the new attribute, click the drop-down list in the Name column and select the name of any option, such as hint.

    That name appears as the new entry in the Name column.

7. Double-click that name to select it.

    Both the name and value blocks are highlighted.

8. Keeping your cursor in the Name block, type over the existing name and enter the new name: icon.

    Double-click inside the value column and enter the path name to the custom image (such as a magnifying glass) to use as a search icon on the form.

    The default value is custom/images/icon.gif, without additional drive or directory specifications.

    To use an alternative image file (*filename*.gif), copy it to the custom/images folder and specify custom/images/*filename*.gif in the value column.

    Note: When users click this icon on the form, an action occurs. This action is specified by the icon_action attribute, which is described in a later step.

9. Save your changes by clicking OK to close the form.

10. Open the form again to continue defining attributes.

11. Select the Employee ID field again, and specify the following new attributes, using the previous steps as a model:

Name=icon_action

icon_action is the fixed attribute name.

Value=*pre-supplied or custom function*

Specifies the JavaScript function to run when the end user completing the form clicks the icon specified by the previous attribute (icon). The JavaScript function is either the ca_PopupVariableWindow(customMain) function or a custom function, as follows:

Value=ca_PopupVariableWindow(customMain)

The ca_PopupVariableWindow(customMain) function prompts the user to enter a value for the key input field. Using that value, the function runs the query for pre-populating the text fields.

In the Employee ID example, the function prompts the user to enter the user ID, and then uses that user ID to run the query for pre-populating the text fields with related personnel data such as first and last names, email address, and so forth.

Value=customCallback(*custom function*)

Specifies the name of the custom JavaScript function to run.

You must use the customCallback function to call your custom function. The customCallback function is included in the custom_form_example.js file.

Specify custom functions in the custom_form_lib.js file or custom_form_example.js file, as explained in <u>JavaScript Functions for Reuse, Customization, and Troubleshooting</u> (see page 240).

Name=icon _tooltip

icon_tooltip is the fixed attribute name.

Value=*tooltip*

Specifies a text string that provides the tool tip text for the icon. When users mouse-over the icon, this text displays above the icon to explain its function.

Name=prefill

> prefill is the fixed attribute name.

Value=*report data object*

> Specifies the id of the report data object that retrieves the data used to pre-fill the text fields on the form.

> Example:
> $report_data.78870b102af20c13425451163b2d2370-7ff41195054288150$

Note: The field attributes named icon, icon_attribute, and icon_tooltip are specific to the CA Service Catalog implementation of CA Workflow and therefore may not be supported in another CA product's implementation of CA Workflow.

12. Click OK to close the form and save your changes.

13. Save the process definition through which you accessed the form (USM_Forms_Container).

14. Customize and test the form, as explained in JavaScript Functions for Reuse, Customization, and Troubleshooting (see page 240).

    If necessary, refine the form, export it again, and test it until it works successfully.

15. Finalize the form so that you can test it in actual requests; for details, see Finalize the Form (see page 229).

Finalize the Form

To finalize the form, preview it and adjust it as needed, and then export it.

To finalize the form

1. Click Preview to see how your form appear to users. Make any necessary adjustments.

2. Click Export.

    The browse dialog appears.

3. Browse to the %USM_HOME%/view/forms folder and save the form with a new file name, such as Personnel Data for Emp Term.xml.

    Important! Ensure that you use a different file name, to prevent your changes from being accidentally overwritten by a default form included in a later patch or upgrade. Keeping the original form intact is also helpful in case you need to verify the default contents of the form later for any reason.

4. Click OK on the Form Designer.

5. Check the Form Designer and verify that your form appears and functions as you intended.

    Note: For troubleshooting help with pre-population and JavaScript functions, see JavaScript Functions for Reuse, Customization, and Troubleshooting (see page 240).

6. If necessary, refine the form, export it again, and verify that it meets your requirements.

    Note: Remember that the Employee ID example, is *not* intended as an end product to "plug into" your system as-is but rather as an illustration to demonstrate how you can modify the default forms to match your organization's unique resources and needs.

    You are ready to include this form in a service or service option group, so that you can test it in an actual Catalog request.

How to Create Groups and Tabs in CA Workflow Forms

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

On a CA Workflow form, you can optionally use controls to organize related fields into logical groups. By default, no such groups are used, and therefore forms appear to end users as one continuous column. Thus, if you have several fields on a form, they typically appear as one long column that users must scroll through. However, instead of using this default setup, you can optionally organize related items on a form into groups. Moreover, when you use groups, you can assign a tab to each group. When users click a tab, only the fields in that group appear on screen, but information that they previously entered is retained. Users do not need to scroll down to view all fields but rather click each tab to view the fields in each group. This section uses a form named New Employee to illustrate how to create and use groups and tabs on existing forms. This process includes the following procedures:

1. Create the first group and organize the input controls in it that appear as fields.

2. Create the second group and organize its input controls.

3. Specify the tab effect for the form.

4. Test the form in a service.

Create the First Group

In this procedure, you create the first group on the form and organize the input controls (fields) that belong to it. For illustration, this procedure uses Name as the group name.

1. Open the form you want to use and decide how to group the fields.

   Note: If you need to create new fields, see How to Pre-Populate Text Fields in Forms (see page 219) and the *CA Workflow IDE Online Help.*

2. On the top left of the tool bar pane, click the Group input control, drag it to the Submit button, and drop it.

   The Group input control appears as follows:

   The new Group element appears under the Submit button and above the New Employee First Name input control, as follows:
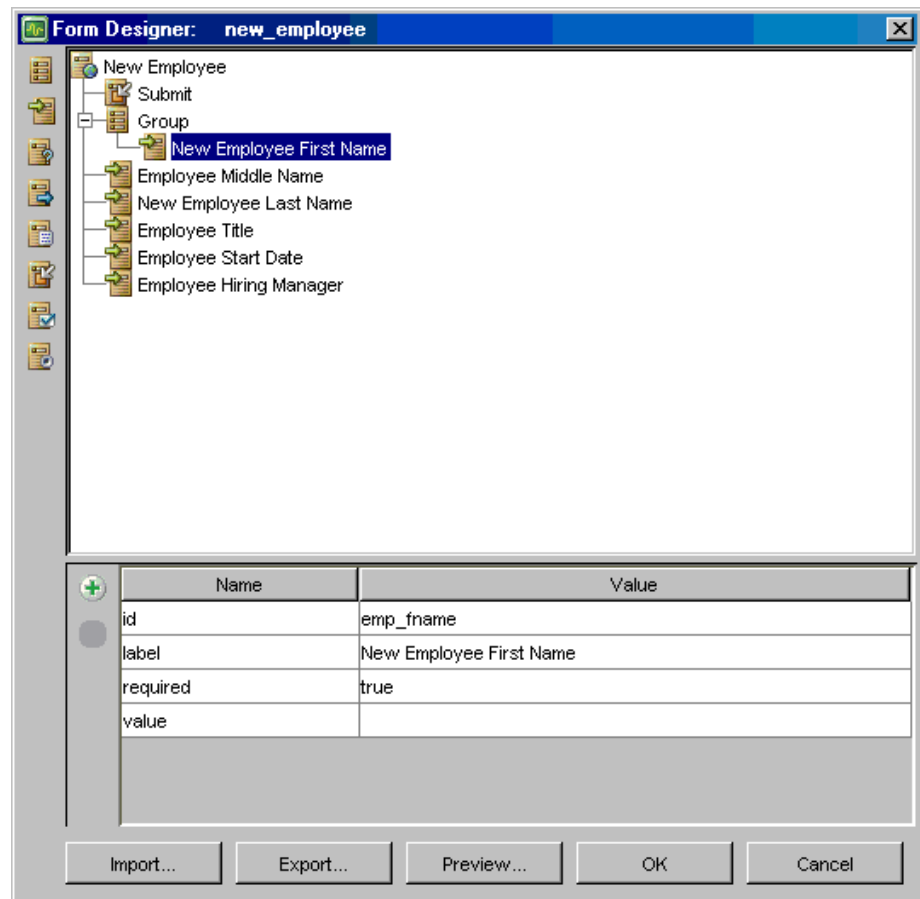
   When users view the form, that input control appears as the New Employee First Name field.

3. Click and drag the New Employee First Name input control to the Group input control, and drop it.

    The New Employee First Name input control appears indented under the Group input control, indicating that the New Employee First Name field appears in the new group on the form.

4. Click and drag the Employee Middle Name input control to the Group input control, and drop it.

    Employee Middle Name appears under New Employee First Name, and both are indented under the Group input control.

5. Click and drag the Employee Last Name input control to the Group input control, and drop it.

    Employee Last Name appears under the other first and middle name input controls, and all are indented under the Group input control.

6. Name the group, as follows:

    a. Select the Group input control.

        The Attributes box displays the attributes for Group.

    b. For the id attribute, click the Value column and enter Name.

        This value becomes the unique ID for this input control.

    c. For the label attribute, click the Value column and enter Name.

        The name of this group becomes Name.

The form designer looks similar to the following:
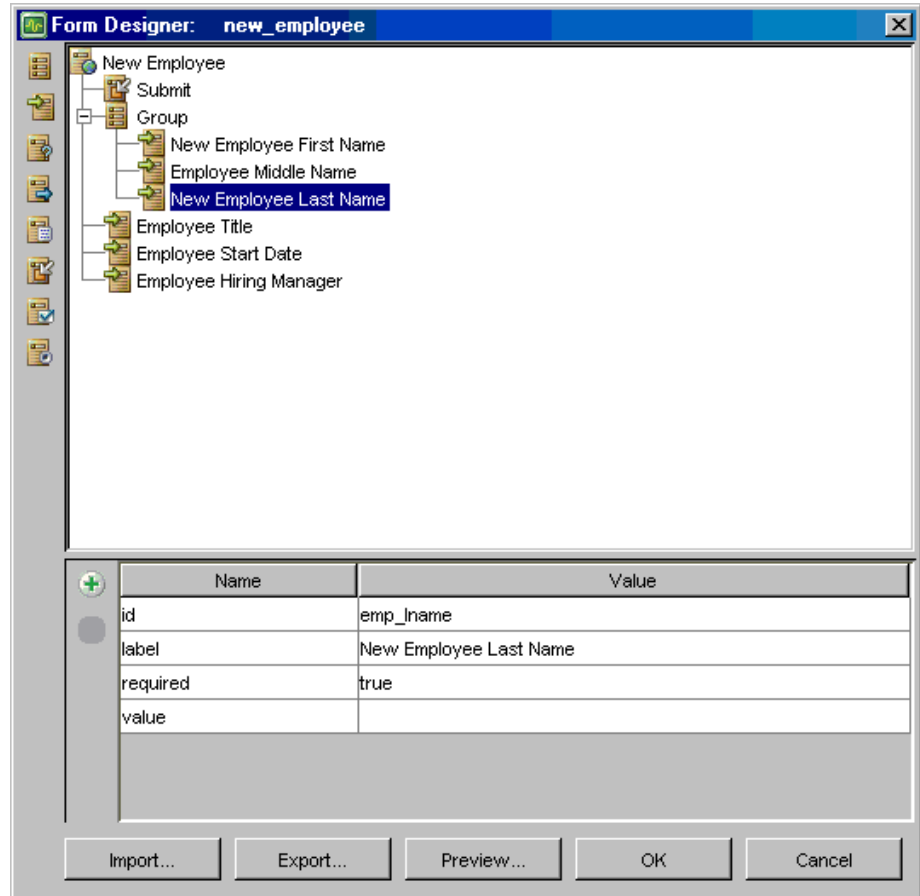
7. Click Preview for a basic idea of how the form appears to users when it is included in a request for a service in CA Service Catalog. Also note how the HTML display reflects the form designer setup in CA Workflow.

    The preview form appears similar to the following, showing that some fields are grouped while others are not:

8. If necessary, make any adjustments to your form; for assistance, see the previous steps or the CA Workflow help.

9. Do the following to save your changes:

    a. Click Export.

        The Save dialog appears.

    b. Enter a unique name for the new form, ensure that the path name meets your specifications, and click Save.

    Note: You may not be able to change the original form; therefore, save the newly modified form as described in this step. As a best practice, save your changes often.

Create the Second Group

In this procedure, you create the second group on the form and organize the input controls (fields) that belong to it. For illustration, this procedure uses Hiring Details as the group name.

1. Click the minus sign next to the Name group to collapse its list of input controls.

2. On the top left of the tool bar pane, click the Group input control, drag it to the Submit button, and drop it.

    The new Group input control appears above the Name Group.

3. Click the Name group, drag it to the Submit button, and drop it.

4. The Name Group appears above the new Group input control.

5. One at a time, click and drag the Employee Title, Employee Start Date, and Hiring Manager input controls to the new Group input control, and drop them. Use the previous related steps as a model, if necessary.

    The Employee Title, Employee Start Date, and Hiring Manager input controls appear indented under the Group input control, indicating that they appear in the new group on the form.

6. Name the group, as follows:

    a. Click the Group input control to select it.

        The Attributes box displays the attributes for Group.

    b. For the id attribute, click the Value column and enter hiring_details.

        This value becomes the unique ID for this input control.

    c. For the label attribute, click the Value column and enter Hiring Details.

        The name of this group becomes Hiring Details.

The form designer looks similar to the following:

7. Click Preview for a basic idea of how the form appears to users when it is included in a request for a service in CA Service Catalog. Also note how the HTML display reflects the form designer setup in CA Workflow.

   The preview form appears similar to the following, showing that all fields are organized into two groups:

8. If necessary, make any adjustments to your form; for assistance, see the previous steps or the CA Workflow help.

9. Save your changes to the form, as explained in previous steps.

Specify the Tab Effect for the Form

In this procedure, you specify the *appearance* attribute to make the names of the groups appear as tabs on the form. When users click a tab, only the fields in that group appear on screen, but information that they previously entered is retained. Users do not need to scroll down to view all fields but rather click each tab to view the fields in each group. This technique makes the form as a whole easier for users to view and process. You need to perform this procedure only once for the entire form, no matter how many groups the form includes.

1. Click the New Employee input control to select it.

   This input control is located at the top of the tree on the form designer.

2. On the field attribute table, click the green plus (+) sign icon next to the Name column.

   A new attribute is created.

3. In the row for the new attribute, click the drop-down list in the Name column and select *appearance*.

    The name *appearance* appears as the new entry in the Name column.

4. Click inside the Value column.

    A drop-down list appears inside the column.

5. Select *minimal*.

6. Save your changes, as described in earlier steps.

7. Click Preview for a basic idea of how the form appears to users when it is included in a request for a service in CA Service Catalog.

    The preview form appears similar to the following, showing that all fields are organized into two groups, each marked with a tab:
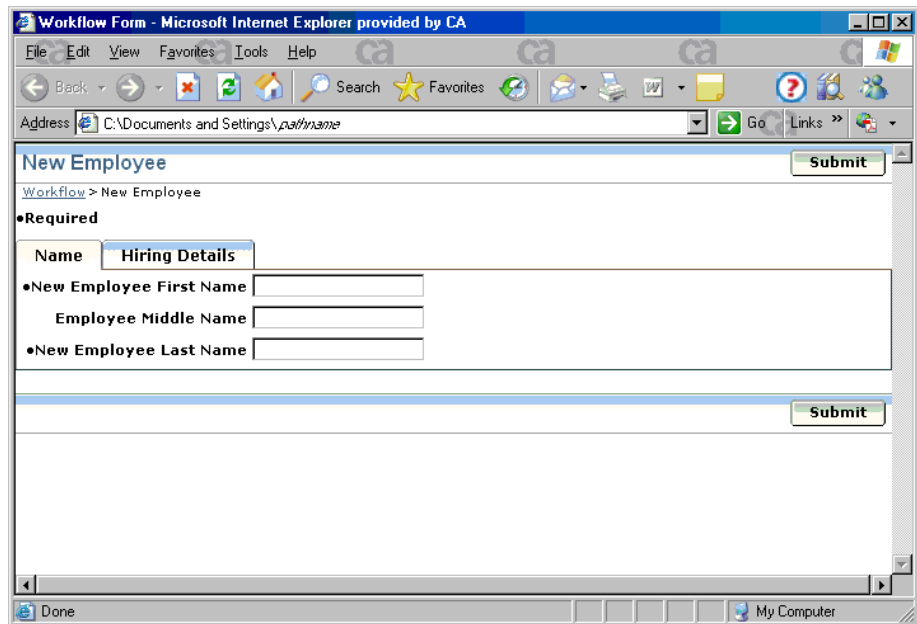
    Note: To enable the tabs to display as shown in the previous screen, you may need to configure your internet browser to allow active content to display. If you do not, both columns may continue to display without the tabs in the preview.

8. Save your changes to the form, as explained in previous steps.

Test the Form in a Request

Test the form in a request in CA Service Catalog, to ensure that it appears to end users as you intend.

1. In CA Workflow, do one of the following:

   - Export the form to %USM_HOME%/view/forms.

     If you export the form to this location, you do not need to upload it from Service Builder.

   - Export the new form to a different location that you can access from Service Builder.

   For details, see the CA Workflow IDE Help.

2. In Service Builder, do the following:

   a. If you exported the form to %USM_HOME%/view/forms, skip this step and go to the next step (b).

      If you exported the form to a different location, upload the form now.

      For details about uploading the form, see Upload a Form (see page 207).

   b. Create a service and service option group in which to include the form.

      For details, see the sections Services (see page 141) *and Service Option Groups* (see page 169).

      Alternatively, optionally use an existing service and service option group, if they meet your criteria.

   c. Attach the form to the service option element that uses the form.

      For details, see Define a Service Option Element (see page 179).

3. As a CA Service Catalog user, complete a request for that service and verify that the form works as intended.

   For details, see Shop from the Catalog (see page 564).

4.   If necessary, refine the form until it works as intended.

The following sample screen shows a sample request containing the New Employee form modified in the previous procedures:
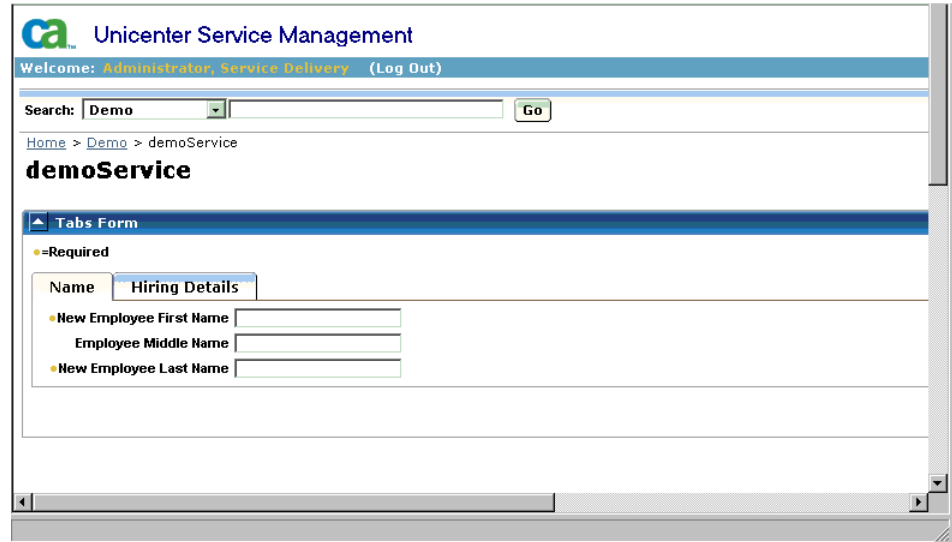
How to Use JavaScript Functions to Customize CA Workflow Forms

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

JavaScript Functions for Reuse, Customization, and Troubleshooting

CA Service Catalog provides JavaScript functions for reuse, customization, and troubleshooting in your custom forms. To use these functions, you must be a proficient user of JavaScript and AJAX. These JavaScript functions are stored in the following files:

■   The ca_form_lib.js file is a JavaScript library that contains CA-specific functions that you can reuse in your forms, for example, to fetch data, call report data objects, and so forth.

This file is located in the default scripts directory, USM_HOME\View\WebApps\USM\Explorer\scripts.

Important! Become familiar with this file but do *not* edit this file directly. If necessary, you can copy and modify functions from this file and use them in either the custom_form_lib.js file or the custom_form_example.js file, which are described next.

■ The custom_form_lib.js file contains a sample function named prefillTextBoxes that you can use as a model for creating your own additional custom library functions. The prefillTextBoxes function retrieves data and uses it to populate the related fields in the form named Employee Termination (EmployeeTermination.xml), as explained in the extended example started in How to Pre-Populate Text Fields in Forms (see page 219).

The custom_form_lib.js file is intended for custom functions that you create and reuse, such as common data validation functions.

To debug new functions that you create in custom_form_lib.js, enable debugging in the custom_form_example.js file, which is described in the next bullet.

The custom_form_lib.js file is located in USM_HOME\View\WebApps\USM\custom\explorer\scripts.

■ The custom_form_example.js file contains sample JavaScript functions. The custom_form_example.js file is intended for main functions with sub functions, such as a control area. You enable and disable debugging in this file, as explained in How to Troubleshoot JavaScript Functions (see page 241).

The custom_form_example.js file is located in USM_HOME\View\WebApps\USM\custom\explorer\scripts.

How to Troubleshoot JavaScript Functions

To code and troubleshoot your custom JavaScript functions, perform these tasks:

1. In the custom_form_example.js file, ensure that the following variable is set to true, as follows:

```
var custom_form_example_on = "true";
```

When this variable is set to *true*, all your custom functions in the custom_form_lib.js and custom_form_example.js files are enabled, as long you use the **customCallback(***custom function***)** to call them when you define attributes for pre-populated text fields (see page 226).

Similarly, when this variable is set to *false*, all the custom functions are disabled, as long as you follow this best practice.

By default, this variable is set to *true*.

2. Code the new function or functions in custom_form_lib.js or custom_form_example.js. Use the JavaScript functions for reuse, customization, and troubleshooting (see page 240).

3. In custom_form_example.js, enable debug mode by setting the custom_debug_on variable to *true*, as follows:

   var custom_debug_on = "true";

   To disable debug mode, set the custom_debug_on variable to *false*.

   Enabling and disabling debug mode does not affect your custom functions unless you use the **customCallback(***custom function***)** to call them when you define attributes for pre-populated text fields (see page 226).

4. Ensure that each function to be debugged has a debug flag coded.

5. Save any changes that you have made to custom_form_lib.js or custom_form_example.js.

6. Optionally clear the cache from your web browser; delete the cookies and clear the history.

   Performing this step is important, because JavaScript updates are often cached.

7. If you performed the previous step, log in to CA Service Catalog again, because you are logged out automatically when you delete the cookies.

8. Using the CA Service Catalog GUI, test the form or forms that use the *.js files that you updated (custom_form_lib.js or custom_form_example.js).

   When debugging is turned on, a Debug Alert confirmation dialog appears before each function is run, and you must click OK to continue.

   Ensure that you test the forms in a service, from request through fulfillment.

9. Record whether each function runs successfully or results in an error.

10. If any function causes an error, check the function, fix the errors, and try running it again. Continue checking, fixing, and testing until the function runs successfully.

    Optionally use the logging features of your web browser, for example, the Error Console feature of Mozilla Firefox.

After the functions run successfully, you have succeeded in customizing your forms.

How to Use Date Controls in CA Workflow Forms

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

Many organizations prefer to use a date (calendar) control to ease the process of collecting date-related information into a form. To add a date control to a form created in CA Workflow, perform the following tasks:

1. Log in to CA Service Catalog as a user with the Service Delivery Administrator role, if you have not done so already.

2. Log in to CA Workflow.

3. In CA Workflow, create or update the form that uses your new query.

   a. Ensure that all required fields are included on the form.

   b. If necessary, create a new field for the date control.

   Use the following the following sources of information:

   ■ For general details about creating and updating forms, see Design a Form to Associate to a Service Option Element (see page 199).

   ■ For specific instructions to add the date control, see Modify a Form to Include Date Controls (see page 244).

4. In CA Workflow, do one of the following:

   ■ Export the form to %USM_HOME%/view/forms.

     If you export the form to this location, you do not need to upload it from Service Builder.

   ■ Export the new form to a different location that you can access from Service Builder.

   For details, see the CA Workflow IDE Help.

5. In Service Builder, do the following:

   a. If you exported the form to %USM_HOME%/view/forms, skip this step and go to the next step (b).

     If you exported the form to a different location, upload the form now.

     For details about uploading the form, see Upload a Form (see page 207).

     b.   Create a service and service option group in which to include the form.

         For details, see the sections Services (see page 141) and Service Option Groups (see page 169).

         Alternatively, optionally use an existing service and service option group, if they meet your criteria.

     c.   Attach the form to the service option element that uses the form.

         For details, see Define a Service Option Element (see page 179).

6.   As a CA Service Catalog user, complete a request for that service and verify that the form works as planned.

    For details, see Shop from the Catalog (see page 564).

7.   If necessary, refine the query, the form, or both until they work as planned.

For details, see the sections previously mentioned in this overview.

Modify a Form to Include Date Controls

When needed, you can select a form and modify it to include date controls, also known as a calendar feature. This feature enables users of a form in a catalog service to specify a date by selecting it from a calendar. As a form designer, you add a date control to a form by modifying a new or existing field to include a new attribute named *show_calendar* with a value of *true*, as shown in this procedure. For illustration, this topic uses the form named Employee Termination (EmployeeTermination.xml). Employee Termination is one of many forms included by default when you install CA Workflow from the CA Service Catalog installation media.

Note: This procedure includes steps to create a new field and add the show_calendar attribute to it, but you can optionally add this attribute to an existing field.

To select a form and modify it to include date controls

1.   If you have not already logged in to CA Service Catalog as a Service Delivery administrator, do so now.

2.   On the home page, select Administration, Tools, CA Workflow.

    The CA Workflow IDE appears.

3.   Click the Process Manager side tab, and click the Process Definitions icon.

    The Process Definitions appear in the main window, displaying the name, short name, modified date, and so forth for all process definitions.

4.   Click the Name column to sort the process definitions alphabetically by name.

5. Find and double-click the process definition named USM_Forms_Container.

   The CA Workflow diagram for this process definition appears in the main window.

   Note: If this container has not been modified since CA Service Catalog was installed, then the container stores only boilerplate elements and therefore is nearly empty.

6. On the bottom portion of the screen, under the CA Workflow diagram, click the Forms tab.

   An empty Forms Designer template appears.

7. Click Add.

   The Forms Designer dialog appears.

8. Click the Import button.

   The Open dialog appears, prompting you to select the location of the form that you want to import.

9. Browse to the folder named %USM_HOME%/view/forms.

10. Click EmployeeTermination.xml and click Open.

11. When asked whether to replace the existing form, click OK.

    The Form Designer: *form name* dialog appears. In this scenario, the Form Designer: emp_termination dialog appears.

    The lines under the form title contain the existing fields and their attributes, in the order that they appear on the form. When you select a field, the names and values of its attributes appear at the bottom of the form, as shown in a later step.

12. On the left tool bar, the CA Workflow form controls appear in a vertical list. Locate the "Input" control.

    This control adds a new text field. Input is typically the second select control in the list. When you mouse-over the Input control on the left tools, the tool-tip text that displays is similar to the following: "Input control to display as a field to enter information." The Input control icon appears as follows:

13. Click and drag the Input icon to the lines that contain the existing controls and field names, and drop the icon between the lines that contain the fields named "Employee Name" and "Employee Manager Name"

A new line named Input appears; it is the new text field in this form. Your screen appears similar to the following:

14. Select the new field named Input.

The names and values of the attributes of the new field appear, in the table under the form lines.

Note: On the Form Designer, when you select any line, the names and values appear for the corresponding control in the table under the form lines.

15. In that table, specify the values for the attributes of the new field.

    Double click each field in the Value column to open the field and enter text, as follows:

    id

    > Specify an intuitive ID, such as Date_Required_By.
    >
    > **Note:** Each element ID in a form must be unique. For more information about IDs, see the CA Workflow IDE Online Help.

    label

    > Specify an intuitive label, such as Date Required By.
    >
    > This text becomes the label that users see when they view this field on the form.

    value

    > Leave this field empty.

16. To create a new attribute for the selected field (Needed By Date), click the green plus (+) sign icon next to the Name column of the field attribute table.

17. In the row for the new attribute, click the drop-down list in the Name column and select the name of any option, such as hint.

    That name appears as the new entry in the Name column.

18. Double-click that name to select it.

    Both the name and value blocks are highlighted.

19. Keeping your cursor in the Name block, type over the existing name and enter the new name: show_calendar.

20. In the Value field for show_calendar, enter true.

21. Save your changes by clicking OK to close the form.

22. Using the previous few steps as a model, use the Input control to create and place any other date controls that you want to add to this form.

    For each new field, specify the id, label, and attributes, as described in the previous step.

    When you are finished, click OK to close the Form Designer dialog and save your changes.

You are now ready to upload the form and test it in a service, as explained in

Attributes for Validating Fields in CA Workflow Forms

This topic applies to forms created and maintained using CA Workflow. Instead of CA Workflow, we recommend that you use the CA Service Catalog Form Designer to create and customize forms (see page 278).

This section explains the attributes that you can use for validating users' input for fields in forms created using CA Workflow. Using these attributes helps ensure that the user input recorded on forms is actually the data that you want and is recorded in the format that you want. For example, you can use these attributes to ensure that phone number fields accept only numeric input, that date fields accept only your company's standard date format, and that credit card fields accept only valid numbers.

Mutually Exclusive Attributes

In most cases, attributes whose names begin with the validate_ prefix are mutually exclusive; that is, you can use *only* one of them in any single field. For example, you can assign either the validate_credit_card_number attribute or the validate_social_security_format attribute to a specific field, but you cannot assign both of these attributes to the field.

There are two exceptions:

- When you specify the validate_data_type=date attribute to a field, you must also specify these attributes to it: validate_date_range and validate_date_format. For details, see Date and Calendar Attributes.

- When you specify input fields for the credit card type and credit card number, if you want to validate them, you must also specify these attributes for the credit card number field: validate_credit_card_number=true and validate_credit_card_type=ca_showDependencyBetweenFields(*dependent field ID-independent field ID* expression). For details, see Credit Card Attributes.

Miscellaneous Attributes

These attributes are available for general use.

validate_phone_number=true|false

Specifies whether to validate the format of the phone number entered by the user in the field whose attributes you are defining.

If you set this value to true, the system validates the criteria above (10-digit minimum and permitted characters) of the phone number but does *not* verify that the phone number is authentic.

This attribute requires 10 digit minimum with optional parentheses, dashes, spaces, and plus signs as separators in any position. This attribute also allows for more digits, to account for international dialing, if needed.

validate_email_address=true|false

Specifies whether to validate the format of the email address entered by the user in the field whose attributes you are defining.

If you set this value to true, the system validates the format (*user-name@domain-name*) but does *not* verify that the user name, domain name, or address is authentic.

validate_social_security_format=nnnnnnnnn|nnn-nn-nnnn|nnn nn nnnn

Specifies whether to validate the format of the social security number entered by the user in the field whose attributes you are defining.

The system validates that there are nine digits entered in the format you specify but does *not* verify that the social security number is authentic.

validate_data_type=number|string|date

number specifies that the user may enter only numbers (0-9) in the field whose attributes you are defining; no letters or other characters are accepted in this field. This specification is useful when no further validation is required for a numeric field. For example, it may be useful in situations when you want to record a quantity or a user's age, weight, or clothing size.

string specifies that the user may enter only letters (a-z or A-Z) in this field; no numbers or other characters are accepted.

date specifies that the user may enter only a date in this field.

Important! The date is validated according to the specifications in the validate_date_range and validate_date_format attributes, which are described next.

When validate_data_type=date, the following attributes are required: validate_date_range and validate_date_format.

Date and Calendar Attributes

validate_data_type=date

As noted above, the following attributes are required when validate_data_type=date. These attributes do not apply otherwise.

validate_date_range=all|future|past

all specifies that the user may enter any date in the field whose attributes you are defining: past, current (the date when the form is being completed), or future.

future specifies that the user may enter may enter the current date or any future date, but not any past date, in the field whose attributes you are defining.

past specifies that the user may enter the current date or any past date, but not any future date, in the field whose attributes you are defining.

Accepts a value of 1-31 days for any month and 1-12 months per year.

If you are uncertain about how many days are in a specific month for a specific year, use the calendar attribute below.

validate_date_format=*format*

Specifies the date format. When completing the form and entering the date in the field whose attributes you are defining, users are required to use the date format you specify, including the dashes or slashes between the values for day, month, and year.

The *format* may be any of the following formats:

■ *M-d-yyyy*

■ *M/d/yyyy*

■ *d/M/yyyy*

■ *d-M-yyyy*

■ *yyyy/M/d*

■ *yyyy-M-d*

M - Specifies the month of the year, such as 1 for January, 4 for April, 10 for October, and so forth. Valid values for *M* are 1-12.

d- Specifies the day of the month. For all months except February, the valid values for *d* are 1 to 30 or 31, depending on the number of days in the specified month. For example: 1-30 for September and November, and 1-31 for July and August. For February, the valid value for *d* are 1-29 for leap years and 1-28 for all other years.

yyyy - Specifies the year; for example, 2009, 2012, 2015, and so forth. Valid values are 1582-4481.

Note: If you specify an invalid or unsupported date format, the default date format *M/d/yyyy* is used.

show_calendar="true|false"

Specifies whether to include a calendar with the field whose attributes you are defining. When you include the calendar, users can click the calendar icon to open a calendar, browse one month at a time, and select a date.

If you use the calendar attribute, it recognizes and enforces the date range and date format selection criteria that you specified in the validate_date_range and validate_date_format attributes.

Credit Card Attributes

These attributes require fields for the user to specify the type of credit card and the credit card number. In this document, for illustration, the input IDs for these fields are named credit_card_type and credit_card_number, respectively.

The valid values for the credit_card_type field are, for example, amex (for American Express), mastercard (for Master Card), and visa (for Visa). Using standard CA Workflow process definition tools for designing forms, you create a drop-down menu to present these options to users.

The valid values for the credit_card_number field are set by the format specifications set by the company issuing the credit card. Each company sets its own format specifications. Using standard CA Workflow process definition tools for designing forms, you create an input field for users to enter their credit card numbers.

Note: For details about using these design tools, see this *Administration Guide* and the CA Workflow help.

To validate that the user's credit card number is valid for the specified credit card type, you must specify the following attributes for the credit_card_number field. These attributes do not apply otherwise.

validate_credit_card_number=true|false

Specifies whether to validate the credit card number entered by the user in the field whose attributes you are defining. Specify true to validate the number or false to skip validating it.

validate_credit_card_type=ca_showDependencyBetweenFields (*dependent field ID-independent field ID expression)*

> You can use this function to validate the user's input for the dependent field against the user's input for the independent field. In this example, we need to validate credit card numbers according to credit card type. Based on the user's selection of credit card type (independent field), you can verify that the credit card number (dependent field) entered by the user is valid. Thus, if the user selects a credit card type of American Express, you can validate that the credit card number entered is valid for American Express. CA Service Catalog uses a special algorithm for the specified credit card type to validate the numbers.
>
> In this example, the complete valid expression is the following:
>
> validate_credit_card_type=ca_showDependencyBetweenFields(document.getElementById('credit_card_nu mber'),document.getElementById('credit_card_type'))
>
> CA Service Catalog validates the format but does *not* verify that the credit card number or user is authentic.
>
> ca_showDependencyBetweenFields(*dependent field ID-independent field ID expression*) is a JavaScript function stored in the ca_form_lib.js JavaScript library. This library and its functions are part of the <u>JavaScript functions for reuse, customization, and troubleshooting</u> (see page 240).

Request SLAs and Calendars

> Request SLAs, outages, outage calendars, and business hours help measure quality of service during the request life cycle.

Request SLA

CA Service Catalog supplies administrators with the ability to create request SLAs to monitor whether service options in a request are processed within the time period that you specify for each monitored state. Your SLAs specify time to warning and time to violation for the selected service option. A single request SLA specifies the amount of time permitted between specified statuses, for example, the time taken to move from Submitted to Approved or from Approved to Completed.

For each request SLA, you specify the starting and ending statuses to monitor; the length of time to reach warning and violation thresholds, the expected level of compliance, and related settings. For details about how to specify SLAs, see How to Create Request SLAs (see page 256).

Setting SLA warning and violation thresholds helps service providers to track the progress of a request. The request SLA data is also stored in the CA Service Catalog database so that it can be included in reports.

The monitoring of time for an SLA warning or violation (SLA time) is started and stopped according the service hours defined by both the outage calendars (see page 255) and the business hours (see page 255) specifications associated to the service. Administrators optionally associate one outage calendar and one business hours specification to each service. During service hours, the SLA time is monitored. During non-service hours, the SLA time is not monitored.

You can optionally specify automated actions to be initiated when a request SLA reaches warning status or violation status. These actions include a pre-defined action for sending email alert messages, as explained in How to Configure Automated Email Alerts for SLA Warnings and Violations (see page 261).

Note: Request service level agreements (SLAs) are a feature of CA Service Catalog, while Quality of Service (QoS) SLAs are available *only* if CA Service Catalog is integrated with CA Business Service Insight. The terms *request SLA* and *QoS SLA* are used when needed in the documentation to distinguish between the two types of SLAs.

Outages, Outage Groups, and Outage Calendars

An outage is a single scheduled event when a service is not available. Examples include individual holidays such as Christmas and New Year's Day (in several countries), Memorial Day (in the United States), Bastille Day (in France), and so forth. When a holiday falls on a weekend, it is typically observed with an outage on either the previous or following weekday. Other outages can include annual or quarterly maintenance or disaster recovery activities.

These holidays and other examples occur annually. However, as needed, you can also specify outages that are one-time only, such as a non-recurring outage to allow for a company to move physical locations or make major hardware or software changes, such as a system-wide database migration.

Outage groups are logical sets of related outages, such as Chinese Holidays, Japanese Holidays, Indian Holidays, or U.S. Holidays. These outage groups typically recur annually. Similarly, if your organization has four outages each year for maintenance, you can include them all in an outage group named Maintenance Outages.

As an administrator, you use outages and outage groups to create *outage calendars*. Outages and outage groups are the major ingredients of an outage calendar. Both outages and outage groups are independent entities that can be reused in several different outage calendars. Outage calendars are associated to services.

For each business unit, administrators, at their discretion, associate *one* outage calendar to a service. The criteria you use for determining the outages, groups, and calendars for each service in a business unit may be based on geographical areas, departments, time zones, other customer-related considerations, or any other groupings that are important in your organization.

Business Hours

Business hours specify the regularly scheduled days and times of service, typically something like one of the following:

- 9:00 a.m. to 5: p.m., Monday through Friday
- 7:00 a.m. to 7:00 p.m. every day

Business hours are set by the service provider, but for best customer relations are geared towards the typical business schedule of all requestors. Thus, a global service provider of urgently needed services, such as emergency medical supplies, may want to maintain 24-7 business hours to meet the needs of all customers. You associate business hours and an outage calendar to a service to define the service hours for the service.

Choose the Service Options to Monitor

CA Service Catalog administrators decide which service options to monitor with request SLAs and which ones to leave un-monitored. The following criteria may help you determine which service options to monitor with SLAs:

■ Check whether any *critical* business rules, business needs, or contractual obligations with service consumers affect any service options and therefore indicate that the service options require SLAs.

■ Check whether it is expected that *all* service options in your implementation are monitored with SLAs.

How to Create Request SLAs

As an administrator, you optionally create request SLAs to monitor whether service options in a service option group are processed within the time period that you specify. Your SLAs specify time to warning and time to violation for the selected service option. To create SLAs, follow this process:

Note: Request service level agreements (SLAs) are a feature of CA Service Catalog, while Quality of Service (QoS) SLAs are available *only* if CA Service Catalog is integrated with CA Business Service Insight. The terms *request SLA* and *QoS SLA* are used when needed in the documentation to distinguish between the two types of SLAs.

1. Select Service Builder, Services.

2. Expand the tree and select the service to which you want to add the SLA.

3. Click the Define icon of the service.

4. Click the plus (+) sign next to the service option group name to expand the tree and display the service options.

5. Locate the service option to which you want to add the SLA.

6. In the Actions column for that service option, click the Set SLA icon.

   The SLA Definition dialog appears.

7. Click Add to create the SLA.

   A new SLA definition appears.

8. For each SLA, specify the following:

    ■ Starting and ending statuses to monitor. A sample starting status is Submitted, and a sample ending status is Approved.

    ■ Length of time, in days, hours, and minutes, to reach warning status.

    ■ Length of time, in days, hours, and minutes, to reach violation status.

    ■ A meaningful descriptive name; for example, a name that summarizes the starting status, ending status, and duration. Examples include Submitted-to-Approval Done--Warning--4 Hours, Submitted-to-Approval Done--Violation--4 Hours, and so forth.

    ■ The Expectation Level, the percentage of SLAs that you expect to be completed without warnings or violations, for example, 80, 90, or 100.

9. Review the projected violation time: to update it, click Project Violation Time. After you change the length of an SLA, click Project Violation Time.

    You can associate an outage calendar (see page 255) and business hours (see page 255) to the service. Both of them (if specified) and the lengths of time that you specify for the SLAs are used calculate the projected violation time. The starting time for this calculation is the current time.

10. If you create multiple SLAs, select one of them as the key SLA by clicking the option button in the Key column. If a service option contains only one SLA, then that SLA is the key SLA.

    When a service option has multiple SLAs, the status of the key SLA determines the status of the service option as a whole, regardless of the status of the other SLAs. For example, suppose you create four SLAs for a service option. If the key SLA has not reached a warning or violation when the request containing the service option is completed, then the SLA for that service option is satisfied, even if all of the other three SLAs *did* reach a warning or violation.

    Conversely, if three of the four SLAs have no warning or violation, but the key SLA has a warning or violation, then the service option as a whole has a warning or violation. With this information in mind, choose the key SLA with special care.

    Both warnings and violations are reflected in SLA Reports.

Note: You can optionally set up automated email alerts to be triggered if an SLA reaches warning or violation status, as explained in How to Configure Automated Email Alerts for SLA Warnings and Violations (see page 261).

How Request SLAs Are Processed

Request SLAs are processed according to both the fixed rules specified by CA Service Catalog and the Maximum Delay for Request SLA Alerts setting that you specify.

Fixed Rules

Request SLAs are processed according to the following fixed rules specified by CA Service Catalog:

■ After a request that contains one or more SLAs is submitted, the SLA clock for each SLA instance starts tracking time when the request reaches the starting status specified for the SLA. For example, if an SLA monitors the period from Pending Approval to Approved, the SLA clock for this SLA instance starts tracking time when the request status changes to Pending Approval.

■ An SLA instance completes when either of the following occur:

− The request reaches the ending status of the SLA before the SLA violation time expires. In this case, the SLA instance completes successfully. The SLA violation time is the maximum length of time permitted for the request reach the ending status of the SLA.

Thus, in the previous example, if the request status changes from Pending Approval (starting status) to Approved (ending status) before the SLA violation time expires, the SLA instance completes successfully.

− The SLA violation time expires before the request reaches the ending status of the SLA. In this case, the SLA instance completes unsuccessfully, causing an SLA violation to occur.

■ The SLA clock tracks time for each active SLA instance and alerts the Catalog system if the SLA warning or violation time expires. The SLA clock is paused and resumed *only* when a CA Service Catalog user holds or resumes (see page 598) the request containing the SLA instance. If a user changes a request with an active SLA instance to Hold status, then the SLA clock is paused. When a user changes such a request to Resume status, the SLA clock is resumed.

Note: If the user of another product integrated with CA Service Catalog puts a request on hold in that product, the CA Service Catalog request, the SLA instance remains active. For example, if CA Service Catalog is integrated with CA Service Desk Manager and a CA Service Desk Manager user holds a CA Service Desk Manager request or change order associated with a CA Service Catalog request that has an active SLA instance, the SLA clock is *not* paused. To pause the SLA clock, you must hold (see page 598) the request in CA Service Catalog.

■ The SLA clock is stopped immediately when a request is canceled.

■ The SLA clock is never restarted (reset to zero) under any circumstances, even if the request is rejected and resubmitted.

For example, consider an SLA named "Approved within 1 Day SLA." This SLA monitors from "Submitted" status to "Approved" status with a one day SLA violation time. In this example, the approval process also follows this request status cycle, in the following order:

1. Submitted

2. Pending Approval

3. Approved

4. Approval Done or Submitted

5. Pending Approval

6. Rejected

In this example, if an end user requests a service, one day passes, and an approver rejects the request, then the SLA "Approved within 1 Day SLA" is violated. After the rejection, if the affected end user updates the request detail and re-submits it, the request approval process is repeated, but the SLA instance is not monitored again because it has already been monitored once and completed with a violation.

■ As illustrated in the previous example, if an SLA instance completes, either successfully or with a violation, that same instance is never monitored again under any circumstances.

■ Once a monitored request SLA instance is violated, the violation is included in SLA-related BusinessObjects Enterprise reports, even if the related request is canceled or rejected after the violation occurs. To exclude such records from a report, customize the report.

Request SLA Processor

The request SLA processor is part of Service View and performs these functions:

■ Listens for status changes for requests that have active SLA instances

■ Sends alert messages when SLA warnings or SLA violations occur

■ Tracks time for the SLA clock

Maximum Delay for Request SLA Alerts

The request SLA processor checks for SLA instances to be processed at these times:

- When the SLA clock for an active SLA instance alerts the Catalog system that the SLA warning or violation time has expired

- When it receives a message of a change in the status of a request with an active SLA instance

- When the time period specified in the Maximum Delay for Request SLA Alerts setting expires. To access this setting, select Administration, Configuration, Request SLA. This setting specifies how frequently the request SLA processor checks for SLA warnings or violations. This setting applies to all SLA instances managed by a Service View computer.

In a Service View-clustered environment, if a clustered computer fails, then event notifications, including SLA alert messages, may be delayed until the failed computer is restored or until other computers in the cluster begin doing the work of the failed computer. Consequently, SLA warning and violation messages may not be issued on time.

To minimize this possible delay in SLA processing time when a Service View- clustered computer fails, you can configure the Maximum Delay for Request SLA Alerts to best meet your needs. The smaller the value you set, the greater the frequency with which the request SLA processor checks the SLA clock for warning and violation times. Therefore, set a less frequent time period, such as one hour, to be informed of SLA warnings and violations quickly. Otherwise, set a larger time period, such as one day, to be notified about SLA warnings and violations later, for example, when the failed clustered computer is restored.

Setting the Maximum Delay for Request SLA Alerts helps reduce any delay of SLA processing in case the clustered computer that started the SLA clock becomes unavailable for a length of time longer than a warning or violation period. In such cases, other active clustered computers take over SLA processing from the failed computer, as soon as any one of the following events occurs:

- The Maximum Delay for Request SLA Alerts time period expires

- An SLA warning or violation is issued

- A status change occurs in a request with an active SLA instance

How to Configure Automated Email Alerts for SLA Warnings and Violations

As an administrator, you may want to create automated actions to be initiated when a request SLA reaches warning status or violation status. Such actions can include using the pre-defined actions for sending email alerts and running your own custom actions, such as running command line commands. Using automated actions may help you find the root cause of a problem sooner and correct it earlier, thereby helping to reduce the number of any additional related SLA warnings or violations.

For information about creating your own custom automated actions for SLAs, such as running command line commands, see Events, Rules, and Actions (see page 26).

To create automated email alerts for SLA warnings and violations, follow this process:

1. Select Home, Administration, Tools, Events-Rules-Actions.

    The Event Types appear.

2. Click the Event Type named Request SLA Alerts.

3. Decide whether to enable one or both of the pre-defined SLA-related rules: they are named When a Request SLA Instance is Warned and When a Request SLA Instance is Violated.

4. Click Enable to enable one or both of these rules.

5. For each rule that you enabled, click Enable to enable the matching action or actions.

    The pre-defined (built-in) action for the SLA instance warning is named Email Alert for Warned Request SLA. Similarly, the pre-defined action for the SLA instance violation is named Email Alert for Violated Request SLA.

    Note: You may also see additional, custom actions that were created by you or other administrators earlier.

6. Click the Edit (Pencil) icon in the Edit column of the action.

    Specify the following to configure the automatically generated email:

    ■   Optionally refine the name and description

    ■   Change the status to Enabled

    ■   In the From Field, enter "Catalog system" or another name that is meaningful in your environment

    ■   In the To Field, enter the email addresses of one or more CA Service Catalog administrators in your environment

    ■   Optionally, change the execution mode or set the timeout value

7. Note that the type is preset to Email and cannot be changed.

8. Similarly, note that the Subject and Message fields are pre-set to provide the following information and cannot be changed:

- The warning or violation notice, including the starting and ending statuses of the request

- The name of the affected service option, service, and request

- The date and time of the warning or violation

- Any applicable subscription information

9. Test the action by submitting a request with at least one SLA and letting the SLA reach warn status, violation status, or both.

10. If necessary, refine the action to meet your requirements.

11. Using the previous steps as a guide, ensure that both actions (Email Alert for Warned Request SLA and Email Alert for Violated Request SLA) are configured as you want them to be.

How to Create and Maintain Outages

You create a scheduled outage to specify a single occurrence when a service is not available.   Reasons for such outages include holidays, maintenance periods, disaster recovery activities, one-time events such as major hardware or software changes. Both outages and outage groups (see page 255) are independent entities that can be reused in multiple outage calendars (see page 255). During the outage periods defined by outage calendars, the monitoring of time for a request SLA (see page 254) is stopped.

To create and maintain outages, follow this process:

1. On the CA Service Catalog GUI, click Service Builder, Service Hours.

    The Service Hours window appears, with Outages selected and in focus.

2. Check the business unit identification message under the main menu and verify that you want to create this outage for the current business unit.

    If Yes, skip to the next step.

    If No, click Change Business Unit (see page 142) and specify the business unit you want.

    After changing the business unit, you return to the Service Hours-Outages window and are ready to create or maintain outages for the newly selected business unit.

3. Do one of the following:

    ■   To create a new outage, click Add.

    ■   To search for one or more existing outages, specify your search criteria in the Search box and click Search. The list of outages displayed is limited to match your search criteria.

    ■   To edit an existing outage, click its Edit (pencil) icon.

    ■   To delete an existing outage, click its Delete (trash can) icon. In this case, the remaining steps in this process do not apply.

    The outage fields appear; required fields are marked.

4. When adding or editing an SLA, complete all required fields, and complete any optional fields that apply. In these fields, supply the following information:

    ■   A meaningful name for the outage, such as Christmas Holiday, New Year's Holiday, Monthly System Maintenance Outage, or Disaster Recovery Setup.

    ■   A meaningful description for the outage, such as annual, monthly, or one-time maintenance procedure.

    ■   The date and time that the outage starts.

    ■   The duration of the outage, in days, hours, and minutes.

    ■   The time zone on which the outage is based, such as GMT+10:00 - Australia/Sydney.

    ■   Whether the outage occurs only once or repeats multiple times (recurring).

■   If the outage is recurring, specify the recurrence pattern, such as yearly, monthly, weekly, daily, or hourly.

■   If the outage is recurring, specify the ending date and time, if applicable.

5.   Click Save.

This outage is ready to be used in an outage group or outage calendar to specify a time period when the monitoring of time for a request SLA is stopped.

How to Create and Maintain Outage Groups

Outage groups are logical sets of related outage events, such as Chinese holidays, Indian holidays, Australian holidays, or annual maintenance days. These outage groups typically recur annually. As an administrator, you combine individual outages into outage groups. Outages and outage groups are the major ingredients of an outage calendar. Both outages and outage groups are independent entities that can be reused in several different outage calendars. During the outage periods defined by outage calendars, the monitoring of time for a request SLA (see page 254) is stopped.

To create and maintain outage groups, follow this process:

1. On the CA Service Catalog GUI, click Service Builder, Service Hours.

    The Service Hours window appears. Under the main menu, Outages is selected in the Menu on the left of the window and is in focus.

2. In that Menu, click Outage Groups.

    Outage Groups is selected and in focus.

3. Check the business unit identification message under the main menu and verify that you want to create this outage group for the current business unit.

    If Yes, skip to the next step.

    If No, click Change Business Unit (see page 142) and specify the business unit you want.

    You return to the Service Hours-Outage Groups window and are ready to create or maintain the outage groups in the newly selected business unit.

4. Do one of the following:

    ■ To create an outage group, click Add.

    ■ To search for one or more existing outage groups, specify your search criteria in the Search box and click Search. The list of outage groups displayed is limited to match your search criteria.

    ■ To maintain an existing outage group, click its Edit (pencil) icon.

    ■ To delete an existing outage group, click its Delete (trash can) icon. In this case, the remaining steps in this process do not apply.

    The fields for the new or existing outage group appear in the Outage Group Detail box; required fields are marked.

    For existing outage groups, any outages that are already included in the group appear in the Associated Outages box.

5. When creating or editing an outage group, do the following:

    a. Complete all required fields and complete any optional fields that apply. In the fields, supply the following information:

        ■ A meaningful name for the outage group, such as U.S. Holidays, Japanese Holidays, or Annual Maintenance Plan.

■   A meaningful description for the outage group, such as annual U.S. holidays or annual maintenance schedule.

    b.   Click Save.

6.   To add an outage to the current outage group, do the following:

    a.   Click Associate Outages.

The Associate Outages dialog appears, listing the outages that you can add to the outage group.

    b.   Optionally, enter search criteria in the Search box and click Search to display only the outages that match your search criteria.

    c.   In the Select column, check each outage that you want to add to the group.

    d.   When you are finished selecting outages, click Associate.

You return to the Service Hours-Outage Groups window. The outages that you selected are now listed as associated to this outage group.

    e.   Repeat the previous steps, as needed, to add more outages to the group.

7.   To remove one or more outages from the current group, do the following:

    a.   In the Select column of the list of associated outages, check the outages that you want to remove.

    b.   When you are finished selecting outages, click Disassociate.

You return to the Service Hours-Outage Groups window. The outages that you selected are no longer listed as associated to this outage group.

This outage group is ready to be used in an outage calendar to specify a time period when the monitoring of time for a request SLA is stopped.

How to Create and Maintain Outage Calendars

As an administrator, you use outages and outage groups to create *outage calendars*. For each business unit, you can optionally apply *one* outage calendar to a service. You cannot apply outages or outage groups to a service directly; they must be associated to a calendar. The criteria you use for determining the outages, outage groups, and calendars for each service in a business unit may be based on geographical areas, departments, time zones, other customer-related considerations, or any other groupings that are important in your organization. Both outages and outage groups are independent entities that can be reused in several different outage calendars.

During the outage periods defined by outage calendars, the monitoring of time for a request SLA (see page 254) is stopped.

To create and maintain outage calendars, follow this process:

1. On the CA Service Catalog GUI, click Service Builder, Service Hours.

    The Service Hours window appears. Under the main menu, Outages is selected in the Menu on the left of the window and is in focus.

2. In that Menu, click Outage Calendars.

    Outage Calendars is selected and in focus.

3. Check the business unit identification message under the main menu and verify that you want to create this outage calendar for the current business unit.

    If Yes, skip to the next step.

    If No, click Change Business Unit (see page 142) and specify the business unit you want.

    You return to the Service Hours-Outage Calendars window and are ready to create or maintain the outage calendars in the newly selected business unit.

4. Do one of the following:

    ■ To create a new outage calendar, click Add.

    ■ To search for one or more existing outage calendars, specify your search criteria in the Search box and click Search. The list of outage calendars displayed is limited to match your search criteria.

    ■ To maintain an existing outage calendar, click the Edit (pencil) icon.

    ■ To delete an existing outage calendar, click the Delete (trash can) icon. In this case, the remaining steps in this process do not apply.

    The fields for the new or existing outage calendar appear in the Outage Calendar Detail box; required fields are marked.

    For existing outage calendars, any outages or outage groups that are already included in the calendar are listed in the Associated Outages box. Both outages and outage groups are listed in this box: for outage groups, the Group Name column is completed; for outages, the Group Name column is empty.

5. When adding or maintaining an outage calendar, do the following:

    a. Complete all required fields, and complete any optional fields that apply. In these fields, supply the following information:

        ■   A meaningful name for the outage calendar, such as Addis Ababa--Level 1 Support Calendar or Rio de Janeiro--Level 3 Retailer Calendar.

        ■   A meaningful description for the outage calendar, such as annual outages for level 1 support group in Addis Ababa or annual outages for level 3 retailers in Rio de Janeiro.

    b. Click Save.

6. To add one or more outages to the current outage calendar, do the following:

    a. Click Associate Outages.

        The Associate Outages dialog appears, listing the outages that you can add to the calendar.

    b. Optionally, to display only the outages that match your search criteria, enter search criteria in the Search box and click Search.

    c. In the Select column, check each outage that you want to add to the calendar.

    d. When you are finished selecting outages, and click Associate.

        You return to the Service Hours-Outage Calendar window. The outages that you selected are now listed as associated to this outage calendar.

    e. Repeat the previous steps, as needed, to add more outages to the calendar.

7. To add one or more outage groups to the current outage calendar, do the following:

    a. Click Associate Outage Groups.

        The Associate Outage Groups dialog appears, listing the outage groups that you can add to the calendar.

    b. Optionally, to display only the outages that match your search criteria, enter search criteria in the Search box and click Search.

    c. In the Select column, check each outage group that you want to add to the calendar.

    d. When you are finished selecting outage groups, and click Associate.

        You return to the Service Hours-Outage Calendar window. The outage groups that you selected are now listed as associated to this outage calendar.

    e. Repeat the previous steps, as needed, to add more outage groups to the calendar.

8. To remove one or more outages or outage groups from the current calendar, do the following in the list of associated outages and outage groups:

    a. Check the ones that you want to remove.

    b. Click Disassociate.

Note: When you select or de-select *one* outage in an outage group, the *entire* outage group is selected or de-selected.

9. Finish your changes and click Done.

   You return to the Service Hours-Outage Calendars window.

After selecting the outages and outage groups for your outage calendar, you can associate the calendar to a service, so that the monitoring of time is stopped for the request SLAs during the outage periods specified in the calendar.

How to Set Up Business Hours

Business hours specify the regularly scheduled days and times of service. Business hours are set by the service provider, but for best customer relations are geared towards the typical business schedule of all requestors.

As an administrator, you can optionally apply *one* outage calendar and *one* business hours specification to a service. As with calendars, the criteria you use for determining the business hours for each service in a business unit may be based on geographical areas, departments, time zones, other customer-related considerations, or any other groupings that are important in your organization.

During the time periods defined by business hours (see page 255), time is monitored for the request SLA (see page 254) attached to a service, except during the outages specified by the outage calendar attached to the service.

To create and maintain business hours specifications, follow this process:

1. On the CA Service Catalog GUI, click Service Builder, Service Hours.

   The Service Hours window appears. Under the main menu, Outage Events is selected in the Menu on the left of the window and is in focus.

2. In that Menu, click Business Hours.

   Business Hours is selected and in focus.

3. Review the business unit identification message under the main menu. Verify that you want to create this business hours object for the current business unit.

   If Yes, skip to the next step.

   If No, click Change Business Unit (see page 142) and specify the business unit you want.

   You return to the Service Hours-Business Hours window. You are ready to create or maintain the business hours specification.

4. Do one of the following:

   ■ To create a new business hours specification, click Add.

   ■ Optionally, to display only the existing business hours specifications that match your search criteria, enter search criteria in the Search box and click Search.

     The list of business hours specifications is limited to match your search criteria.

   ■ To update an existing business hours specification, click the Edit (pencil) icon.

   ■ To delete an existing business hours specification, click the Delete (trash can) icon. In this case, the remaining steps in this process do not apply.

   The fields for the new or existing business hours specification appear; required fields are marked.

   Any existing business hours specifications are listed.

5. Complete all required fields, and complete any optional fields that apply. In these fields, supply the following information:

   ■ A meaningful name, such as Addis Ababa--Level 1 Business Hours or Rio de Janeiro--Level 3 Retailer   Business hours.

   ■ A meaningful description, such as in-service hours for level 1 support group in Addis Ababa or in-service hours for level 3 retailers in Rio de Janeiro.

   ■ The day of the week and length of time for service. For each day of the week, specify the entire day, no service at all, or a custom-defined time period when service is to be available.

   ■ The time zone on which these business hours are based, such as GMT-8:00 Pacific Time (US&Canada) or GMT+10:00 (Australia/Sydney).

6. Click Save.

7. As needed, repeat the previous steps to create or modify additional business hours specifications.

After you select and save your business hours specification, you can associate it to a service.

How to Associate Outage Calendar and Business Hours to a Service

The outage calendar (see page 255) and business hours (see page 255) object associated to a service work together to specify the time periods during which the service is expected to be available. During these periods, the time is monitored for the Request SLAs (see page 254) attached to the service.

You are not required to associate an outage calendar and business hours object to a service at the same time. However, we recommend that you do so, to help ensure that the Request SLAs attached to the service are governed by established, visible periods of expected up time (business hours) and down time (outage calendars). For example, if you attach a Request SLA to a service without specifying either an outage calendar or business hours, then clients may expect that the service is to be available 24 hours per day, 365 days per year, which is rarely the case.

To associate outage calendars and business hours to a service, follow this process:

1. Click Service Builder, Services.

    The Services window appears, in tree form.

2. Expand the tree to access the service you want.

3. Under the Action icons, click the Outage Calendar and Business Hours icon. The default icon looks like a calendar and its tool tip text is "Outage Calendar and Business Hours."

    The Associate Outage Calendar or Business Hours dialog appears. This dialog displays the currently associated outage calendar and business hours specification, if applicable.

    This dialog also displays Show|Hide toggle switches for searching and selecting each specification. Similarly, this dialog displays Use Default buttons that are activated only when applicable.

    You can optionally associate outage calendars and business hours to the service, or you can restore the default selections for each one, as explained in the next steps.

4. To associate an outage calendar to the service, do the following:

    a. Click Show Calendar Search.

        The list of available outage calendars appears. The currently associated calendar (if applicable) is selected.

    b. Optionally click Search to limit the calendars displayed to match your search criteria.

    c. Select the calendar that you want and click Associate.

        The list of available outage calendars disappears, and the calendar you selected is shown in the Associated Outage Calendar field.

5. To associate business hours to the service, do the following:

   a. Click Show Business Hour Search.

      The list of available Business Hours specifications appears. The current association (if applicable) is selected.

   b. Optionally click Search to limit the business hours to match your search criteria.

   c. Select the business hours that you want and click Associate.

      The list of available business hours disappears, and the business hours you selected is shown in the Associated Business Hours field.

6. Business unit administrators can optionally specify a default outage calendar and business hours for all services in their business units (see page 273). If a service is associated to a different outage calendar, you can optionally assign the default outage calendar by clicking the calendar's Use Default button. Similarly, if a service is associated to a non-default business hours, you can optionally assign the default business hours by clicking the business hours' Use Default button.

   Note: If the business unit has no default outage calendar specified, then clicking Use Default results in the service being associated to no outage calendar, meaning that time is tracked for SLAs associated to the service every day of the year. Similarly, if the business unit has no default business hours specified, then clicking Use Default results in the service being associated to no business hours, meaning that time is tracked for SLAs associated to the service every hour of the day.

7. Click Close.

   You return to the Service Builder, Services page.

The outage calendar and business hours that you selected are associated to the service.

How to Specify Default Outage Calendar and Business Hours

Business unit administrators can optionally specify a default outage calendar and default business hours for all services in their business units. Doing so helps ensure consistency in the days and hours that services are available throughout the business unit.

If the business unit administrator assigns no default outage calendar and if the administrator for a service does not associate a custom outage calendar to the service, then time is tracked for SLAs associated to the service *every day of the year*. Similarly, if the business unit administrator assigns no default business hours and if the administrator for a service does not associate custom business hours to the service, then time is tracked for SLAs associated to the service *every hour of the day*.

Therefore, as a best practice, business unit administrators and administrators of individual services work together. Together, they verify that the default outage calendar and default business hours for the business unit are reasonable and achievable. Otherwise, avoidable SLA warnings and violations may be issued for such services.

To specify a default outage calendar and default business hours for all services in a business unit, follow this process:

1. Review the business units that need to have default outage calendar and default business hours specified. For each of these business units, review the outage calendars and business hours that have been created already, and identify the ones that you want to use as defaults. Check the details of your selections carefully to verify that they meet your requirements for all services in the business unit. If no suitable options exist yet, create them, as explained in How to Create and Maintain Outage Calendars (see page 267) and How to Set Up Business Hours (see page 269).

2. Click Service Builder, Services.

    The Services window appears. The business unit that you are logged in to appears at the top of the screen, with its services listed under it.

3. Verify that you are logged into the business unit for which you want to specify the default outage calendar and default business hours.

4. If applicable and if permitted by your role, do the following to display the child business units of the business unit that you are logged in to, so that you can update their default settings:

    a. Click Change Business Unit (see page 142), and select the child business unit you want to display, if applicable.

        Under the business unit that you originally logged into, the child business unit and its services are listed.

    b. Repeat the previous step (a) for each child business unit that you want to update.

        Each business unit name appears on the left and has an expand-collapse icon next to it to expand or collapse its list of services.

5. Scroll to the line that identifies the business unit you want and click Edit.

   The Folder Details dialog for the business unit appears.

6. In the drop-down lists provided, select the default outage calendar and the default business hours.

7. If applicable, repeat the previous two steps to specify the default outage calendar and default business hours for each child business unit, as needed.

   Note: You must set the default outage calendar and default business hours for each business unit *individually*. Child business units do *not* inherit these default settings from their parent business unit.

You can review SLA reports for the affected services to verify that the default settings meet your requirements for the business unit.

How to Save SLA History

At various times, you may need to change one or more of your SLA definitions. One likely reason is a change in service agreement or contract that arises after the monitoring of SLAs has already started. To handle such cases, CA Service Catalog enables you to save your current SLAs as history and reset SLAs. SLA history is *not* accessible using the GUI but *is* reflected in SLA reports.

For example, suppose that you created SLAs for a service on the first day of the month but later in the month needed to change them because of an update in a related contract. At the end of the month, if you create an SLA report for the entire month, the report shows the different warning and violation settings used for the SLAs before and after the changes you made.

When you change SLA settings for a service option, the change takes effect for *future* requests that include the service in which the service option resides. Requests that are in-progress are *not* affected.

To save SLA history and reset SLAs, follow this process:

1. Select Service Builder, Services.

2. Expand the tree and select the service to which you want to add the SLA.

3. Click the Define icon of the service.

4. Click the plus (+) sign next to the service option group name to expand the tree and display the service options.

5. Locate the service option to which you want to add the SLA.

6. In the Actions column for that service option, click the Set SLA icon.

    The SLA Definition dialog appears.

7. Click Save as History.

    Note: This button is enabled *only* if the defined SLA exists.

8. When prompted, specify a reason for saving as history and resetting the SLAs, for example, "need to shorten warning and violation times, per change in contract."

9. Confirm the action.

    The existing SLAs are made inactive. Inactive SLAs are saved for use in both reports and in-progress requests; however, inactive SLAs are *not* applied to any future requests.

    Note: Inactive SLAs are removed from the service and are no longer accessible through the CA Service Catalog GUI, so they are no longer visible on screen.

10. As needed, create request SLAs (see page 256) for use in future requests, to replace the inactive SLAs.

Note: Inactive SLAs are reflected in SLA reports for related requests.

SLA Reports

In both the Report Builder (see page 97) and BusinessObjects Enterprise, you can create, configure, and view reports concerning SLA-related data. This data covers both current SLAs and inactive SLAs that have been saved as history (see page 275).

Keep in mind the following when you check SLA reports for warnings and violations:

- If a key SLA is warned or violated, then the service option associated with that SLA is warned or violated.

- If any service option in a service is warned or violated, then the service containing that service option is warned or violated.

- If any service in a request is warned or violated, then that request is warned or violated.

- For a violation, the Whole Violated Mark appears, at both the service and request levels.

Note: Request service level agreements (SLAs) are a feature of CA Service Catalog, while Quality of Service (QoS) SLAs are available *only* if CA Service Catalog is integrated with CA Business Service Insight. The terms *request SLA* and *QoS SLA* are used when needed in the documentation to distinguish between the two types of SLAs.

Chapter 8: Using the Form Designer

This section contains the following topics:

Introduction to the Form Designer

The Form Designer provides a means to create and maintain forms from within CA Service Catalog. Form Designer forms and CA Workflow forms can be used in the same service.

For details about CA Workflow, including its form design features, see the chapter "Using Service Builder" in this guide and the *Integration Guide*.

How to Create, Customize, and Use Forms

On a new or existing form, you can add and configure elements such as text fields, text areas with check boxes or radio buttons, and so forth. Each element of the form captures information needed to approve and complete a request for a service.

To add and configure elements on a form, follow this process:

1. Become familiar with the Form Designer by reviewing the main components (see page 279) and elements of a form (see page 281). These provide an overview of the main features of the Form Designer and its intended use in the request life cycle.

2. Verify that elements do *not* duplicate information requested through other forms used for the same service, including system forms (see page 377). System forms request standard information such as name, address, phone number, and so forth for shipping purposes.

3. Decide which elements you want on the form and in which order. As a best practice, create a "rough draft" of the form on paper or using a graphics application. Ensure that each part of the form has a matching element on the Form Designer.

4. Using the Form Designer, create a form (see page 296).

   Alternatively, you may decide to do one of the following instead of using the Form Designer to create a form:

   ■ Use an existing Form Designer form as-is

   ■ Use an existing Form Designer form as a starting point, and modify it to include the changes you want

   ■ Import a CA Workflow form, use it as a starting point in the Form Designer, and modify it to include the changes you want

   Note: If you have an existing CA Workflow form that does not require any changes, you can continue to use it in services as you have done previously.

5. Add the elements (see page 281) you want to the form, one at a time, in the order that you want them to appear.

6. For each element, specify the HTML attributes and JavaScript attributes (see page 301), as follows:

   a. Set the required HTML attributes (see page 303).

   b. Set any optional HTML attributes (see page 303).

   c. Set any JavaScript attributes (see page 315) you want.

7. Check and refine the layout and elements of the form to help ensure that they appear and function as you want them to. If necessary, add, move, and delete elements on the form (see definition on page 298) to meet your requirements.

8. If necessary, if the form is not to be available to *all* business units, assign the form to a specific business unit (see page 375).

9. Attach the form to a service option (see page 366) and test it in a service.

Main Components

You access the Form Designer by selecting Service Builder, Form Designer. The main components of the Form Designer are the Component Tree (the left pane), the Preview Area (the middle pane), and the Item Inspector (the right pane).

When you select a form in the Component Tree, the form appears in preview form in the Preview Area, and the Item Inspector displays the attributes of the form object, as shown in the following sample page:

Figure 1: The Component Tree, Preview Area, and Item Inspector are the major components of the Form Designer
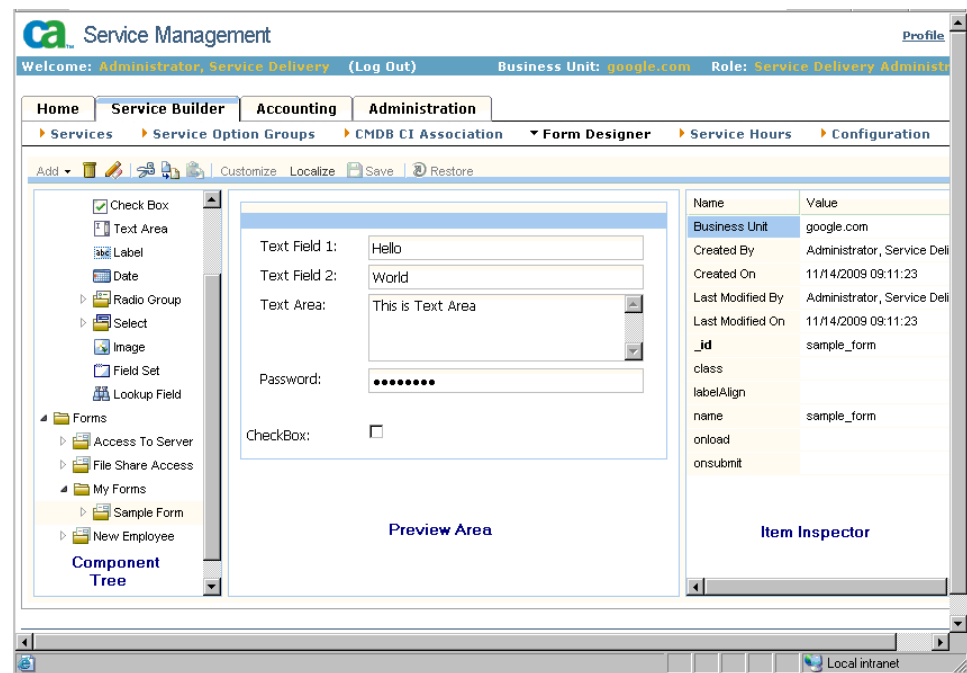
Similarly, if you select an element of a form in the Component Tree (such as Text Field 1 of Sample Form in the previous page), the selected element is highlighted in the form in the Preview Area, and the attributes of the selected element appear in the Item Inspector.

Component Tree

The Component Tree (the left pane) lists all the folders, forms, and form elements in the system in a tree representation. The System folder of the tree contains the elements--such as text fields, select groups, and check boxes--that you can copy and modify to construct the elements of your forms. You can store your forms in a separate folder, for example, the My Forms folder, which includes a sample form to help you get started. In the Component Tree, you can quickly create move and duplicate of existing forms and elements, using the cut, copy, and paste functions of the Form Designer.

Note: You assign access to forms according to business unit. When a form is owned by a parent business unit, all its child business units (also named sub-business units) can access it.

Preview Area

The Preview Area (the middle pane) displays the currently selected folder, form, or form element. The Preview Area enables you to preview the form as a whole, providing you a critical end-user perspective as you add, delete, and modify elements on the form. When you add, delete, or re-order elements, when you modify the attributes of an element that affects the end-user view, the Preview Area immediately shows the change and saves it.

Item Inspector

The Item Inspector (the right pane) displays HTML and JavaScript attributes for the selected form or the selected form element. Required attributes appear in **bold**. For example, in the previous screen, the form named Sample Form is selected in the Component Tree, and the form attributes appear in the Item Inspector. In this example, the *id* attribute is bold because it is required.

Elements of a Form

The Form Designer provides the following elements for you to create and modify forms:

- Basic elements of a form (see page 281)
- Advanced elements of a form
  - Select boxes and options (see page 287)
  - Lookup field (see page 287)
  - Date time field (see page 284)
  - Dual list (see page 286)
  - Static table (see page 288)
  - Dynamic table (see page 290)

You can reuse one or more elements that you have customized, by copying and pasting the elements from one form to another.

Basic Elements of a Form

You can use the following basic elements when you create and maintain forms:

Column Layouts

Use column layouts to organize the components on the form for maximum efficiency. The column layout includes two vertical columns, Column 1 and Column 2, aligned next to each other. You can place form components in one or both of these columns. You cannot insert one column inside another. However, you can place multiple column layouts in a form.

Text fields

Use a text field to enable the user to enter free-form text information to be used by the program, when the information is so customized that presenting a list of choices is impractical or inappropriate. Examples may include fields for the user to enter a name, address, telephone number, or multiple-digit numeric entries. Use text fields when only one line of input is required.

Text areas

Text areas are for the same general purpose as text fields when more than one line of input may be required, for example, a brief answer to an essay question, such as a free-form description of personal habits, characteristics, goals, or ideals.

Label

A label identifies a form or an area of the form. For example, you can use a label at the top of the form to provide the title, such as Medical History Form. Within the form, you may have other labels, such as Family History, Eating Habits, and Illnesses. If you use a multiple-column format in your form, it may be helpful to use a label as a heading for each column.

Use the HTML attribute (see page 303) named Label Text to configure the label that users see on the form, for example, as a title or heading, as noted previously.

You can optionally perform the following tasks related to labels:

■ Use the HTML attribute (see page 303) named Hidden to hide the label when the conditions you specify are met.

■ Use the JavaScript attribute (see page 315) named onClick to run a JavaScript function (see page 342) when the user clicks the label.

Check boxes

Use a check box to enable a user to select or decline a single option. Typically, users select the option to indicate Yes or True, and clear this option to indicate No or False. Examples include check boxes for options such as the following: Add me to your mailing list, Call me to set up an appointment, or I understand and agree to these terms of use.

You can optionally use multiple check boxes to group related choices, of which users may want to select two or more. For example, you can add one check box each for the peripheral devices associated with a laptop computer, such as mouse, docking station, carrying case, external backup drive, and so forth.

Radio groups and radio buttons

A radio group contains radio buttons *only*. A radio group cannot contain any other element. Use a radio group and its radio buttons to present a list of options, from which the user must choose only one. For example, you can create a radio group named Size with three buttons: one each for Small, Medium, and Large.

In contrast to options in select boxes (see page 287), radio group buttons *always* appear on the form. Therefore, to save space on the form, you can use a radio group only for fields having four or fewer options.

Images

Use an image to provide a picture representing an item that is or may be included in an associated service or service option. The image you specify must reside in the filestore, the central location for files used by CA Service Catalog. Use the following format to specify the path name of the image:

FileStore/*path*/*filename.ext*

Important! The folder name *FileStore* is case sensitive. Therefore, use the correct case in path names and all other references.

Note: For details about setting up the filestore, see the *Implementation Guide*.

Field Sets

Use field sets to group multiple elements as a set. Examples include groups of accessories for a laptop (such as battery charger, carrying case, docking station, and so forth) or group of telephone numbers for a user (such as home, work, cell, and fax). Elements in a field set have a box drawn around them to show that they are related.

Spinner field

Enables a user to do the following: select a numerical value from a range of incremented values, such as 100, 200, 300, and so forth.

The user clicks the up or down arrows to increase or decrement the selected value.

Use the HTML attributes for spinner fields only (see page 303) to configure the contents and behavior of the spinner field.

Date Time Field

You can use the Data Time Field (an advanced element) when you create and maintain forms.

Date Time Field

As an administrator, you can set the format of the date and time, using the Date Time field.

The date and time that appear on the Form Designer match the date and time of the browser that you are using to access CA Service Catalog. Similarly, the date and time that appear to other users accessing the form match the date and time of their browsers. Therefore, the date and time of form settings—even on the same forms—may differ between users, especially users in different time zones. This principle applies whether users access the form through the Form Designer or while requesting a service.

Date

Use the date portion of this field whenever date information is required, for example, Start Date, End Date, Date Required, Estimated Date of Arrival, Date Received, and so forth. When users click the date field, a calendar appears and they can select a date, using the month and year tabs to move quickly forwards or backwards from the current month or year.

By default, the date format for the Date Time field matches the date format of the current business unit.

Note: As an administrator, you can optionally change the default format for the business unit to a different format by editing the profile of the business unit on the Administration, Business Units page and selecting a new date format. Examples include M/d/yyyy, d-M-yyyy, or yyyy/M/d.

Specify date formats in Date Time fields on your forms according to the following rules:

Letter	Meaning	Format	Example
y	year	Number	2009
M	month in year	Text or Number	July or 07
d	day in month	Number	10

CA Service Catalog supports the date format attributes supplied with the Google Web Toolkit (GWT) 1.6. For more information about these attributes, see the Google website, www.google.com.

Time

Use the time portion of the Date Time field whenever time information is required, for example, Start Time, End Time, Estimated Time of Arrival, and so forth. When you use the time portion of this field, a drop-down list of time values appears next to the calendar. Users can select a time from this list, using the scroll bar on the list to move quickly to a time earlier or later than the current time.

By default, the time format for the Date Time field matches the time format of the business unit of the user.

Note: As an administrator, you can optionally change the default format for the business unit to a different format by editing the profile of the business unit on the Administration, Business Units page and selecting a new time format.

Specify time formats in Date Time fields on your forms according to the following rules:

Format	Separator
HH:mm	colon (:)
HH.mm	period (.)

If the time portion of the field is hidden; that is, if the Hide Time attribute is set to *true*, then this time format is ignored, and the drop-down list for selecting the time does not appear on the form.

Otherwise, the field accepts valid values separated by a space. CA Service Catalog supports the time format attributes supplied with the Google Web Toolkit (GWT) 1.6. For more information about these attributes, see the Google web site, www.google.com.

For example, you can optionally add literal values enclosed in quotation marks (such as "hours" or "minutes") before or after the hour (HH) and minute (mm) values.

Use the HTML attributes (see page 303) named *hidetime* and *increment* to control whether and how incremented time values appear in each Date Time field on a form.

Dual List

You can use the dual list (an advanced element) when you create and maintain forms.

Displays two columns in a box: The left column lists available options, and the right column lists the selected options.

A toolbar of arrows appears between the columns. Users select and clear options by highlighting them and clicking the arrows to move the options from one column to another. For example, to select an option, you highlight it in the left column and click the arrow to move the option to the right column.

Users can highlight multiple options in one column and click the arrow to move them as a group to the other column. Moreover, users can click the double arrows to move all options from one column to another: This action selects or deselects *all* options.

While you are designing the form, you can perform the same actions as the user to select and clear options.

By default, the dual list field contains *no* available options. To populate the dual list with available options, use *one* of the following methods:

■ Specify a report data object (see page 99) for the dual list.

   To do so, enter the HTML attributes (see page 303) named Report/Plug-in Id and Report/Plug-in Variables.

■ Create a static list of individual options for the dual list.

   To do so, add options to the dual list. You can add, move, and delete (see definition on page 298) options to the dual list. The default dual list element in the Component Tree includes one option. You can copy and modify this option to create additional options.

To enable users to specify the order of the selected options, specify a value of *true* for the HTML attribute named ordered. When you do so, the toolbar is updated to include up and down arrows. Users can highlight selected options and click these arrows to move the options up or down in the list.

If the value of the ordered attribute is *false*, then the up and down arrows do not appear, and users cannot order the selected attributes.

While you are designing the form, you can perform the same actions as the user to order the selected options (in the right column). The available options (in the left column) always appear in the order that you arrange them on the Component Tree.

Lookup Field

You can use the lookup field (an advanced element) when you create and maintain forms:

Lookup field

Use a lookup field and the JavaScript function named ca_fdDoFieldLookup(fieldId, reportId) to populate fields based on user input to a report data object (see page 351).

You configure the lookup field to prompt the user to enter the data, such as a user ID, asset ID, or city, used in the query of the data object. The data object queries your data source, based on the user input to the lookup field, and uses the results to populate the fields you specify.

When the user clicks the associated icon (magnifying glass) for the lookup field and enters the requested data, the query runs and returns the search results in rows. Each matching value appears in a row. Users review the rows and select the row they want. When the user selects a row, the results from that row populate the matching fields on the form.

For example, the lookup field may prompt the user to enter a user ID. The query then searches the database for the first name and last name of that user ID and presents them to the user in the search results. The user can select one row in the search results to populate the corresponding First Name and Last Name fields on the form.

Select Boxes and Options

Select boxes and options

A select box contains select options *only*. A select box cannot contain any other element. Use the default setting of a select box and its options to present a list of choices, from which the user must select only one. For example, for a shoe order form, you can create a select box named Width with three options: one each for Narrow, Standard, and Wide.

The options for select boxes appear on the form only when the user clicks the drop-down list box to display the options. Therefore, you can use a select box rather than an option group to save space on a form. Doing so is especially relevant when the number of options is four or more.

Moreover, you can optionally change the default setting of a select box to enable users to select two or more options instead of only one. To do so, change the Multi-Select attribute of the select box from False (the default) to True. If you change this setting to True, the appearance of the select box changes from a drop-down list to a sequential list or list box, meaning that all options are always displayed and therefore require more space on the screen. The user sees multiple lines in the select box and can select multiple options.

When this attribute is set to false, the select box appears as a combo box on the form: The user sees a list box and can select only one item from the list. Catalog users cannot enter a *custom* value in a select box. However, users can type inside a select box. When they do so, the drop-down list is populated with the options that begin with the typed text. Users can select the option they want from this "auto-complete" list.

To set specifications for the auto-complete list, use the Minimum Search Characters attribute, one of the HTML attributes for select boxes only (see page 308). Use these attributes to configure the appearance of options in this list, including pagination.

Create a Static Table

You can create a static table to enter structured data into a form. A static table is a type of container, like a field set, that can contain certain elements of a form. You can use the columns in the table to organize the data from each type of element. In contrast to a dynamic table (see page 290), a static table consists of fixed data that you specify manually.

Follow these steps:

1. Edit or create the form in which you want to add the table. Expand the form.

2. Do the following to add the table element to the form:

   a. (Optional) Create a field set to contain the table.

   b. Expand the System folder, drag the Table element, and drop it on the form. If applicable, drop the table onto the field set that you created in the previous step.

   c. Specify an _id value for the table and save the form.

3. Do the following to add an element to the table:

   a. Expand the table to display the Row field.

   b. Drag the element (see page 281) that you want from the System folder and drop it on the Row field. You can drag-and-drop the following elements:

      ■ Date Time field

      ■ Label

- Select field whose Multi-Select attribute is set to false. This setting allows only a *single* selection.

- Spinner

- Text

The name of the element that you drag-and-drop becomes the name of the first column. For example, if you drag-and-drop a Date element, the name of the first column becomes Date.

Similarly, the data that you enter in the column must match its element. For example, in a Date column, you can enter only dates.

c. Specify an _id value for the column and save the form.

Note: After you save the form, you can optionally rename the element that you dragged and dropped. If you rename the element, the name of the column changes accordingly. For example, if you rename the element to Start Date, the name of the column also changes to Start Date.

d. Configure each element that you add to a table, the same as if you were adding the element to the form without a table. Each one is a <u>basic element</u> (see page 281), except the <u>Date Time field</u> (see page 284) and the <u>Select field</u> (see page 287).

For date fields the value returned must be a long or appropriately formatted string. For label columns the value is converted to a string. For spinner columns, the value must be an integer or double. For text columns, the value is converted to a string.

4. Repeat the previous step for each element that you want to add to the table.

The second element that you add specifies the name and data type of the second column, as illustrated in the previous step. For example, you can add a Text element as the second column. In that case, the second column is titled Text and must contain text data. As noted in the previous step, you can optionally rename the element, which automatically renames the column.

5. To add rows to the table, do the following:

a. Select the Row field on the Table element on the System folder.

b. Drag-and-drop it on the Table element on the form.

c. Repeat these steps until you have added all the rows that you want.

Note: You cannot move, copy, cut, or paste rows.

6. Define values for each row in the table, as follows:

    a. In the first row, specify the static values of each cell, using their value attribute.

    b. In the remaining rows, specify values using the column attribute.

    The Form Designer does *not* validate any actual data or the format of any data that you enter in the rows of the table. However, when users display the form in a request, the Catalog system validates the data and displays it *only* if you use the correct format. Thus, any invalid values that you specify do *not* appear when users view the form in a request. For example, if you specify a string value for a date column, the corresponding table cell appears empty to the user. Therefore, be careful to specify the correct format. For example, enter only dates in Date Time fields and only text in Text fields.

    c. Perform this step if applicable; otherwise, skip it.

    If you are using a Select field (see page 287) whose Multi-Select attribute is set to false, then the Select field does *not* contain the value attribute. In that case, do the following:

    ■ For the first row: Enter the value of the Selected Index attribute of the Select field. For example, to specify the first option, enter 0. To specify the second option, specify 1, and so on.

    ■ For the remaining rows: Copy the value of the *value* attribute from the select option of the Select field. Paste this value into the column attribute of the row.

7. Specify any or all of the following additional attributes:

    ■ HTML attributes (see page 303)

    ■ Attributes for tables only (see page 311)

    ■ JavaScript functions for tables only (see page 349)

You have created a static table.

How to Create a Dynamic Table

You can create a dynamic table to enter structured data from a report data object into a form. Like a static table (see page 288), a dynamic table is also a type of container that can contain certain elements of a form. You can use the columns in the table to organize the data from each type of element. Unlike a static table, however, you populate a dynamic table with a report data object. The report data object can use either a data source query or an API plug-in (see page 381). Follow this process:

1. Create the dynamic table (see page 291).

2. Configure the dynamic table (see page 293).

Create the Dynamic Table

You can create a dynamic table to enter structured data from a report data object into a form.

Follow these steps:

1. Create or edit the report data object or API plug-in (see page 381) that you plan to use to populate the dynamic table.

2. Keep in mind the following: The data returned by the variables in the report data object or API plug-in must match the format required by the columns in the table. Otherwise, the data is not displayed to users when they open the form in a request. Later steps explain this requirement in greater detail.

3. Edit or create the form in which you want to add the table. Expand the form.

4. Do the following to add the Table element to the form:

   a. (Optional) Create a field set to contain the table.

   b. Expand the System folder, drag the Table element, and drop it on the form. If applicable, drop the table onto the field set that you created in the previous step.

   c. Specify an _id value for the table and save the form.

5. Do the following to add the report data object or API plug-in to the table:

   a. Select the Table element.

      The attributes of the Table element appear.

   b. If you are using an API plug-in, specify the values of the following attributes:

      ■ Report/Plug-in Id: Enter the ID of the API plug-in that you want to use. You can find values for these attributes on the Administration, Tools, Plug-ins page. Copy the Id of the plug-in that you want from that page and paste it into the value of the Report/Plug-in Id attribute.

      ■ Report/Plug-in Variables: If applicable, open the API plug-in that you selected to display its details, including variables. On the details page, the Inputs section lists the ID values and descriptions of the input variables for the plug-in. Copy the Ids of the variables that you want from that page and paste them into the value of the Report/Plug-in Variables attribute.

c. If you are using a report data object, specify the values of the following attributes:

■ Report/Plug-in Id: Enter the ID of the report data object that you want to use. You can find values for these attributes on the Administration, Report Builder, Data Objects page. Click the Edit icon of the report data object you want to display its properties. Copy the ID of the report data object from that page and paste it into the value of the Report/Plug-in Id attribute.

■ Report/Plug-in Variables: If applicable, Click the Edit icon for the report data object that you selected to display its properties, including variables. On the properties page, the input variables for the report data object appear as follows:

For a Query: The input variables appear as %*expression*% statements.

For Plug-in: The input variables appear in the Arguments field.

For CSV: The input variables do not apply.

Copy the variables that you want from that page and paste them into the value of the Report/Plug-in Variables attribute.

For both attributes, enter variables as a JSON expression, for example:

$({'<variable name>' : '<variable value>', ...})

$({'userid':_.user.id,'rm_orgunit':ca_fdGetSelectedOptionValues(ca_fd.formId,'orgunit_id')})

Important! Specify variables carefully. If you specify *no* variables, unpredictable results can occur.

d. Save the form.

When users complete this form while requesting a service, the report data object or API plug-in runs and returns the data that you specified.

You have created the dynamic table. Next, you it.

Configure the Dynamic Table

After you create a dynamic table, you configure it to contain the data and use the format that you require.

Follow these steps:

1. Review the following requirements for adding elements to the table. Each element that you add becomes a column in the table.

   ■ Add one element for each variable that you specified in the Report/Plug-in Variables attribute in the previous step.

   ■ (API plug-ins) The value of the _id attribute of each element (column) in the table must match a value of an Output Id of the plug-in.

   ■ (Report data objects) The value of the _id attribute of each element (column) in the table must match a value of a Field of the data object.

2. Do the following to add an element to the table, according to the requirements in the previous step:

   a. Expand the table to display the Row field.

   b. Drag the element you want from the System folder and drop it on the Row field. You can drag-and-drop the following elements:

      ■ Date Time field

      ■ Label

      ■ Spinner

      ■ Text

■ Select field whose Multi-Select attribute is set to False. This setting allows only a *single* selection.

c. Perform this step if applicable; otherwise, skip it.

If you are using a Select field (see page 287) whose Multi-Select attribute is set to false, then the Select field does *not* contain the value attribute. In that case, you can populate the Select field using *either* a static list *or* a report data object.

To populate the Select field *using a static list,* do the following:

■ For the first row: Enter the value of the Selected Index attribute of the Select field. For example, to specify the first option, enter 0. To specify the second option, specify 1, and so on.

■ For the remaining rows: Copy the value of the *value* attribute from the select option of the Select field. Paste this value into the column attribute of the row.

The name of the element that you drag-and-drop becomes the name of the first column. For example, if you drag-and-drop a Date element, the name of the first column becomes Date.

Note: You cannot move, copy, cut, or paste rows.

d. Specify an _id value for the column and save the form.

Important! The _id value for the column must meet the requirements noted in the previous step. Also, the data type and data format must also be the same. Otherwise, the column is not populated with data.

For example, you are using an API plug-in, and you want a column of the table to specify the start date of an event. You drag-and-drop a Date Time element onto the table and specify an _id value of start_date. This element becomes a column in the table. Therefore, the Output Id of a variable of the API plug-in must also be start_date. The sequence does not matter. Moreover, this variable must also return the required date and time data in the format that matches the Date Time element.

Note: After you save the form, you can optionally rename the element that you dragged and dropped. If you rename the element, the name of the column changes accordingly. For example, if you rename the element to Start Date, the name of the column also changes to Start Date.

3. Configure each element that you add to a table, the same as if you were adding the element to the form without a table.

Each one is a basic element (see page 281), except the Date Time field (see page 284) and the Select field (see page 287). The data that you enter in the column must match its element. For example, in a Date column, you can enter only dates.

4. Repeat the previous steps for each element that you want to add to the table.

   Important! All requirements noted in the previous step apply.

   The second element that you add specifies the name and data type of the second column, as illustrated in the previous step. For example, you can add a Text element as the second column. In that case, the second column is titled Text and must contain text data. As noted in the previous step, you can optionally rename the element, which automatically renames the column.

5. (Optional) Do the following to specify a custom value for pagination for the table:

   a. Specify the value of the Page Size attribute of the Table element.

   b. Do one of the following, whichever applies:

      ■ For API plug-ins, configure sorting and pagination parameters (see page 385).

      ■ For report data objects, no additional steps are necessary.

6. (API plug-ins only) Do the following to enable users to sort the results returned in the table.

   a. Specify a value of True for the Sortable attribute of the Table element.

   b. Configure sorting and pagination parameters (see page 385).

   Note: Sorting does not apply to report data objects.

7. Specify any or all of the following additional attributes:

   ■ HTML attributes (see page 303)

   ■ Attributes for tables only (see page 311)

   ■ JavaScript functions for tables only (see page 349)

You have configured the dynamic table.

Create a Form

You may need to create a form to obtain information required for fulfilling a service request, if you are not already using a system form to obtain that information.

To create a form

1. Click Service Builder, Form Designer.

    The Form Designer window appears; the folders are collapsed in the Component Tree (the left pane) and the Form Design Area (the middle pane) blank.

2. In the Component Tree, click the icon next to Components folder to expand it.

    The top level of folders appears in the Component Tree.

3. Expand the sub-folders under the System folder.

    These folders contain the elements that you use to construct a form.

4. Click the icon next to the Forms folder to expand it.

    The My Forms and System folders appear.

5. Click the My Forms folder to select it.

6. Above the Components folder, click the Add button and select Form from the drop-down menu.

7. The Add Form dialog appears.

8. Enter the name of the new form and click OK. The name must be unique.

    The new form is created; its name appears in the My Forms folder. The form attributes appear in the Item Inspector (the right pane).

9. Specify the required and optional form attributes (see page 297).

    Note: These attributes are for the *form itself*. These attributes are *not* the same as the HTML and JavaScript attributes that you specify for the *elements* (such as columns, text fields, and select groups) that you use to create the fields on the form.

    The Created By, Created On, and other form attributes are filled in automatically.

You are now ready to begin adding elements to the form.

Form Attributes

Form attributes apply to the *entire* form rather than to one field only.

name

Specifies the name of the form.

Specify this attribute for each form, and verify that this attribute is unique for each form.

_id

Specifies the identifier of the form.

class

Specifies the CSS class name.

Form Type

Specifies whether the form is a configuration form type.

Valid values are as follows:

- Request – for a request of a service from the catalog
- Configuration – for a content configuration form

Default: Request

Note: This attribute *cannot* be written using JavaScript expression.

Label Align

Applies to labels *only*. This attribute specifies the alignment of the label in relation to the fields on the form.

Valid values are top, right, and left (default). If no valid value is specified, the default is used.

Label Width

Specifies the width in pixels of the field labels in the form.

Note: You can also specify Label Width as an HTML attribute (see page 303) for the column layouts and field sets on the form.

onLoad

Specifies the JavaScript function to run when the user opens the form in a request.

When multiple forms having an onLoad attribute appear on a single page, each JavaScript function (see page 341) runs when the user opens the form that includes the function.

An alert appears if an onLoad function fails to run correctly, for example, if the function cannot be found. If you receive an alert, verify that function is correctly defined and referenced, and that the function runs without errors.

Note: As a best practice, store all custom JavaScript functions in the custom_form_lib.js file in the filestore. For information about the filestore, see the *Implementation Guide.*

onSubmit

Specifies the JavaScript function to run when the user submits the request that contains the form.

This function must return a Boolean value. If the function returns any value except *true*, the form is not submitted.

When multiple forms having an onSubmit attribute appear on a single page, each JavaScript function runs when the user submits the form that includes the function.

The information about alerts for the onLoad function also applies to the onSubmit function.

How to Add, Move, and Delete Elements

On a new or existing form, you can add and configure the elements (see page 281), such as text fields, text areas with check boxes or radio buttons, and so forth. You add and configure elements when you create and customize forms (see page 278).

To add and configure elements on a form, follow this process:

1. In the Component Tree (the left pane of the Form Designer), verify that the Component, System folder is expanded, and that the element you want to add appears in the tree view under the System folder.

    For example, if you want to add a text field, verify that the System folder (under the Components folder) appears and is expanded to show the Text Field option. Similarly, if you want to add a select box and its options, verify that the Radio Group folder (under the System folder) appears and is expanded to show the Radio option.

2. Verify that the form to which you want to add the element already exists. If necessary, create a form (see page 296).

3. Expand the Forms, My Forms folder and scroll to the form to which you want to add the element.

4. Verify that both the element you want to add and the intended form are visible in the tree view at the same time.

5. Click the element, for example, Text Field, drag it, and drop it on the intended form.

   For example, if the intended form is named Model Options, click Text Field, drag it to Model Options, and drop it there, all on the same tree.

   When you click and drag an element, both a checked circle and the message "1 object selected" appear next to the cursor.

   As you drag the mouse, when the dragged element is over a form or existing form element on which it can be dropped, the checked circle turns green, and the cursor changes to appear as a hand. Conversely, the checked circle turns red when the dragged element is not over a form or form element on which it can be dropped.

6. Release the mouse to drop the element on the form or form element. When you drop it, the checked circle must be green. Otherwise, the element is not added.

   Similarly, the element is not added if the attempted action violates a rule. Examples include attempting to drop a select option onto a radio group or to drop a select group onto another select group.

7. Specify the *required* HTML attributes (see page 303) for the new element, and click Save.

   The required HTML attributes for each element appear in **bold**.

   Note: Attempts to perform a different action *before* saving this element fail. For example, attempts to drag and drop a new element *before* saving this element fail.

8. Check the form in the Preview Area (the middle pane) to verify that the element was added to the form as you intended. If it was not, you may want to either rearrange the elements or delete the new element and try adding it again.

9. Specify the optional HTML attributes (see page 303) (if any) that you want for the new element, and click Save.

10. By default, components are organized in one column. To configure the form to use two columns, do the following:

    a. Drag Column Layout from the System folder and drop it onto a desired form.

    b. When you drop this layout, the tree updates with Column Layout and its two children, Column 1 and Column 2 automatically.

    c. Click the icon next to Column Layout to display the two columns, which are named Column1 and Column 2.

    d. Drag and drop other elements, such as text fields, radio groups, and select boxes, onto Column 1 and Column 2, to achieve the layout you want.

Note: When you highlight the Column Layout or a Column, the corresponding areas in the Preview Area of form may be rendered gray and disabled, and the elements within the column may be highlighted with red border.

11. Specify the JavaScript attributes (see page 315) (if any) that you want for the new element, and click Save.

12. To specify a meaningful name for the new element, do the following:

    a. Select the new element in the Component Tree and click the Rename button at top of the Component Tree.

       Note: Verify that you select the new element (such as Text Area or Radio Group) on *your* form in the Component Tree; be careful that you do *not* select the same-named element on the *System* folder in the Component Tree.

       The Name dialog appears.

    b. Enter the new name and click OK.

    For example, in a form named External Storage Drive Order Form, you may drag and drop a radio group and change its name from the default Radio Group to Capacity in GB. Next, you may drag and drop three radio buttons onto that radio group, and change their names from the default radio button to Small (50), Medium (100), and Large (150), respectively.

    Note: When you create a new element, a colon (:) is automatically included at the end of its name. The colon does not appear in the Name dialog when you rename the element. However, it does appear in the Preview Area after you click OK to close the dialog. Verify that you do not enter an additional colon in the Name dialog; otherwise, the element has *two* colons after its name in the Preview Area.

13. Verify that the new element appears and functions as you intended. If necessary, refine any attributes to obtain the appearance and function you want.

14. Repeat the previous steps for each element that you want to add.

15. Optionally, move one or more elements on the form by dragging and dropping them onto a different location in the form, for example, after an element that they previously preceded.

16. Optionally, delete elements you no longer want on the form by selecting them and clicking Remove.

HTML and JavaScript Attributes for Elements

After you add an element to a form, specify (see page 301) the values of its required HTML attributes and any optional HTML attributes or JavaScript attributes. Elements generally have one or two required HTML attributes and several optional HTML and JavaScript attributes.

The names of required attributes appear in **bold** in the Item Inspector when you select the element. Different types of elements have different required HTML attributes. For example, most elements require that you set the value of the _id attribute.

The names of all JavaScript attributes begin with the prefix *on*, such as onClick, or onMouseOver. These attributes specify the JavaScript function to invoke when the user of the form performs an action. For example, for onClick, the action occurs when the user clicks the element. Similarly, for onMouseOver, the action occurs when the user hovers the mouse over the attribute.

Specify Attributes for an Element

After you add an element to a form, specify the values of its required HTML attributes and any optional HTML attributes or JavaScript attributes. Specifying these attributes helps you customize the element to function according to your requirements.

Follow these steps:

1. Click Service Builder, Form Designer.

   The Form Designer appears. The Components and Forms folders are collapsed in the Component Tree (the left pane of the Form Designer). The Preview Area (the middle pane of the Form Designer) is blank.

2. Open your form. For example, expand the Forms, My Forms folder and select the form whose elements you want to configure.

   The selected form appears in the Preview Area.

3. Click the icon next to the form name to expand the folder for the form and display its existing elements.

4. If you have not yet added all the elements you want to the form, then add elements to the form.

5. Select the element whose attributes you want to set.

    The Item Inspector (the right pane) displays the names and values of all HTML attributes and JavaScript functions available for the selected element. The names of required HTML attributes appear in **bold**.

6. In the Item Inspector, search the Name column and locate the attribute you want to set, one of the following types:

    ■ HTML Attributes (see page 303)

    ■ JavaScript Attributes (see page 315)

7. Move your cursor to the adjacent field in the Value column and click.

    The cell you clicked is highlighted and opened for editing.

8. Enter the value for this attribute and click Save.

    For example, set the value for the _id attribute of a Text Field element. You can enter something like "Quantity Field," "Age Field," or "Model Field."

    Your changes are saved.

9. At minimum, verify that all required attributes are set with the correct values.

You have specified the value of HTML attributes and JavaScript attributes.

HTML Attributes

You can use HTML attributes to configure the appearance and behavior of the elements of a form (see page 281). Keep in mind the following principles:

- Not all HTML attributes apply to all elements (fields). As you view a form in the Form Designer, click an element to see which attributes apply to it.

- Required HTML attributes appear in **bold** in the Item Inspector. When you add an element to a form, complete the required attributes and save the form immediately after adding the element. Afterwards, you can add optional HTML attributes and JavaScript attributes.

- The HTML attributes are primarily internal, in that they regulate the appearance and behavior of the form. They *typically* do not supply literal text that appears to the administrator in the Design Area or to the user in a request.

- You view and modify the names and values of the HTML attributes in the Item Inspector as needed. However, you generally do not see the changes reflected as literal text in the Design Area. The exceptions are the Checked attribute for radio buttons and check boxes only, and the Value attribute for labels, text fields, and text areas only.

- For an HTML attribute, you can optionally enter a JavaScript expression, but you cannot call a JavaScript function.

- For all elements except labels and images, you supply the literal text that appears to end users, as follows: You rename the elements on the form in the Component Tree and verify their appearance in the Preview Area. In this way, you add and configure elements on a form (see page 298).

You can specify attributes according to the following categories:

Attributes for most or all elements (see page 304)

Attributes for Date Time fields only (see page 306)

Attributes for field sets only (see page 307)

Attributes for labels only (see page 307)

Attributes for select boxes only (see page 308)

Attributes for spinner fields only (see page 310)

Attributes for tables only (see page 311)

Attributes for text fields and text areas only (see page 314)

Attributes for Most or All Elements

Some HTML attributes (see page 303) apply to *all or most* elements (fields) of a form. The following are the attributes of that group that may require explanation:

_id

Specifies a unique identifier for the form field.

If the _id is not unique, a warning message appears.

Important! The Form Designer *warns* you if the _id attribute values are duplicated for multiple fields but does *not* prevent you from saving them. Therefore, change any duplicate _id attributes a soon as you become aware of them. If the form contains duplicate _id attributes, JavaScript functions can run on the wrong form field and cause errors. If you change the value of the _id attribute, verify that you also change any references to it, especially in JavaScript functions.

CSS Class

Specifies a space-separated list of CSS classes. This class is associated with style information.

Empty Text

Provides additional description of a field to the user.

The value that you specify for this attribute appears inside the field when the user opens the form. This text disappears when the user begins entering a value in the field. A sample value follows, for a password field:

AAbb1234

You can optionally use the Empty Text, Hint, and Tooltip attributes together to assist the user.

Name

Specifies the name of the element.

Value

Specifies the value of the input component. This attribute is often useful for entering sample data or instructions. For example, for the value attribute of a text field named Address, you can specify something like "123 my street" or "Enter your address here." In both examples, the value suggests that users replace the default text with their actual addresses.

Disabled

Specifies a value of true or false.

Hidden

Specifies whether to hide a field, including the label. Specify true or false. When the value is set to true, the field is *not* visible.

You can use JavaScript expressions to specify this value as a condition, as follows: If the condition is met, hide the field. Otherwise, display the field.

For example, you can hide a fulfillment-related field before the request reaches the Pending Fulfillment status. Similarly, you can hide a salary-related field from any user other than the manager of the user and the Human Resources user group.

You can specify all the same conditions (see page 461) that apply for the Policy Builder.

Hide Label

Specifies whether to hide *only* the name of a field. The field remains visible regardless of whether you name the label. This attribute provides flexibility when space is a concern.

Note: The Hide Label attribute does *not* apply to the Label element. Instead, the Hide Label attribute applies to the *name* of certain other fields that you enter in the Component Tree of the Form Designer. After you enter the name of a field in the Component Tree, the name appears in the Preview Pane.

Specify true or false. When the value is set to true, the name for this field is *not* visible.

You can use JavaScript expressions to specify this value as a condition, as follows: If the condition is met, hide the name. Otherwise, display the name. You can specify all the conditions (see page 461) that apply for the Policy Builder.

For example, you can hide the names of obvious fields, such as Name, Address, and Telephone Number, after the user submits the request. Doing so can be helpful to save space on the form when the additional fields become visible for approval and fulfillment statuses only.

Hint

Specifies helper text for a field. This text *always* appears below the field, regardless of the position of the mouse.

For example, for a password field, you can specify a hint similar to the following: "Passwords must be six to eight characters and must include both letters and numbers."

You can optionally use the Empty Text, Hint, and Tooltip attributes together to assist the user.

Required

Specifies whether this field is required. Enter a value of true or false.

When a user submits the form, the system verifies that all the required fields are not empty. A required field is a field whose required attribute is set to true. If any required fields are empty, the system prompts the user to complete them.

Style

Specifies associated style information.

Tab Index

Specifies the position in the tabbing order. Enter the numeric value only.

Tool Tip

Specifies helpful text appears *only* when the user hovers the mouse over the field.

You can optionally use the Empty Text, Hint, and Tool Tip attributes together to assist the user.

Text Direction

Specifies the supported values for left-to-right and right-to-left (ltr/rtl) for the direction of the text.

Attributes for Date Time Field Only

The following HTML attributes (see page 303) apply to the Date Time field (see page 284) *only*.

Hide Time

Applies *only* to the time portion of the Date Time field; this field is an optional element of a form (see page 281).

Specifies whether to display the drop-down list for users to select the time. Specify *true* to hide this list or *false* to display it.

Time Increment

Applies *only* to the time portion of the Date Time field; this field is an optional element of a form (see page 281).

Specifies the increment (in minutes) of the times that appear in the drop-down list for users to select the time. Valid values are any positive integer greater than zero.

For example, if you specify 15, the values that appear in the drop-down list for the user to select include 9:00, 9:15, 9:30, 9:45, 10:00, and so forth. Similarly, if you specify 30, the values that appear in this list include 9:00, 9:30, 10:00, 10:30, 11:00, and so forth.

Attributes for Field Sets Only

The following HTML attributes (see page 303) apply to field sets *only*. Fields sets are basic elements of a form (see page 281).

Collapsed

Applies *only* to the field set element.

Specifies whether the label of the field set is initially collapsed (true) or expanded (false).

Label Width

Specifies the width in pixels of the column layout and field set.

Note: You can also specify Label Width as a form attribute (see page 297).

Attributes for Labels Only

The following HTML attributes (see page 303) apply to labels *only*. Labels are basic elements of a form (see page 281).

Label Text

Specifies the text for a label.

A label identifies a form or an area of the form. For example, you can use a label at the top of the form to provide the title, such as Medical History Form. Within the form, you may have other labels, such as Family History, Eating Habits, and Illnesses. If you use a multiple-column format in your form, it may be helpful to use a label as a heading for each column.

The Label Text attribute of a label contains the text that users see on the form, for example, as a title or heading, as noted previously. This Label Text attribute can contain HTML tags, which are especially helpful for highlighting purposes. For example, to specify the Family History label in bold, enter the following in the Label Text field: Family History Form.

Attributes for Radio Groups Only

The following HTML attributes (see page 303) apply to radio groups *only*. Radio groups are basic elements of a form (see page 281).

Orientation

Lets the user control the orientation of the options for the group.

Valid values are horizontal and vertical (default). If no valid value is specified, the default is used.

Attributes for Select Boxes Only

The following <u>HTML attributes</u> (see page 303) apply to <u>select boxes and options</u> (see page 287) only:

List Width

Specifies the minimum width of the drop-down list for a select field.

The Form Designer uses the *larger* of the following values:

■ The width of the select field

■ The value specified for the List Width attribute

Multi-Select

Specifies a value of true or false.

When this attribute is set to true, the select box appears as a list box on the form, meaning that the user sees multiple lines in the select box and can select multiple options.

When this attribute is set to false, the select box appears as a combo box on the form. The user can select one item from the list. The user cannot enter a custom value for this attribute.

Minimum Search Characters

Applies *only* when the value of the Multi-Select attribute of the select box is False. When this value is False, the select box appears as a combo box on the form: The user sees a list box and can select only one item from the list. Catalog users cannot enter a *custom* value in a select box. However, users can type inside a select box. When they do so, the drop-down list is populated with the options that begin with the typed text. Users can select the option they want from this "auto-complete" list.

The auto-complete list starts to populate after the user types the nth character, where n is the value you set for Minimum Search Characters attribute. For example, if you specify a value of 7, the auto complete options appear after the user types the seventh character.

Selected Index

Specifies the default selection.

The number you enter becomes the number of the default selection. For example, suppose the select box contains 10 values. If you specify 0 for the selectedIndex attribute, the first value in the list becomes the default. Similarly, if you specify 5 for the selectedIndex attribute, the fourth value in the list becomes the default.

You can use a select box in a static table (see page 288).

Report/Plug-in Id

The value is the ID of the report data object (data object) used to pre-populate a combo box (see page 320). The data object that returns two fields:

The first field returns the list of values in the combo box. Users can select one or more of these values, depending on the setting of the multiple attribute. The second field returns the label for the combo box.

The data object represents a stand-alone MDB query that does not require a JavaScript function to run.

Report/Plug-in Variables

Specify this attribute to use user input to pre-populate a select box (see page 322).

For example, you can specify this attribute to use the selection in one select box to determine the values populated in a second select box.

Report/Plug-in Id and Report/Plug-in Variables also apply to tables

Page Size

Specifies the number of options that appear on one page in a drop-down list. The drop-down list appears when a catalog user does one of the following:

■   Clicks the arrow of a drop-down list

■   Types text in a field and thereby activates the auto-complete feature

The Page size attribute applies *only* to the following:

■   Select boxes whose Multi-Select attribute is set to False.

■   Static table (see page 288)

■   Dynamic table (see page 290)

The Page size attribute is optional and accepts *only* positive integers.

This attribute is useful when report object returns too many options to list on a single page. In such cases, the drop down box contains a scroll bar and page numbers. To view the values in the list, users can either scroll or click a page number.

Attributes for Spinner Fields Only

The following HTML attributes (see page 303) apply to spinner fields *only.* Spinner fields are basic elements of a form (see page 281).

This field enables users to select one of many values by clicking the up or down arrow. Each time the user clicks an arrow, the selected value increases or decreases by the amount specified by the Increment attribute.

Typically, the values for a spinner field appear in meaningful, equal increments, such as 100, 200, 300, and so forth.

Allow Decimals

Specifies whether to use decimal values, for example, in the rate or cost.

Allow Negative Numbers

Specifies whether to use negative values.

Increment

Specifies the amount by which the selected value increases or decreases each time the user clicks the up or down arrow. For example, suppose you want to use the following values: 100, 200, 300, and 400. In this example, the increment is 100. If the selected value is 200 and the user clicks the up arrow, the selected value increases to 300.

Minimum Value *and* Maximum Value

Specify the minimum and maximum values, respectively.

In the example for the previous attribute (Increment), the minimum value is 100 and the maximum value is 400.

Number Format

Specifies the format of the number that appears to the user, for example, 1, 2, 3, and so forth.

The Form Designer uses the number format of the Google Web Toolkit. For details, see the Google Web Toolkit website. At publication time, the website is www.google-web-toolkit.googlecode.com.

Attributes for Tables Only

The HTML attributes (see page 303) in this topic apply to tables *only*.

Table elements have several attributes. Table elements have the following types of attributes:

■   Table attributes apply to the entire table, that is, to *all rows* in the table.

■   Row attributes apply only to the selected row.

Descriptions follow for each type.

Attributes for the Entire Table

The following table attributes require explanation:

Report/Plug-in Variables and Report/Plug-in Variables

Apply to dynamic tables (see page 290).

Allow Inserts

Specifies whether users can add rows to tables in a form when they are completing a request.

When this value is false, users *cannot* add rows.

When this value is true, users *can* add rows. For example, you can design a service that contains a table with several predefined options. You can also enable the user to add custom values to the table. The advantage of using such a table over a text box is that the table stores the data in *structured* format.

When the value of Allow Inserts is true, the following conditions exist when users edit the form while requesting a service:

■   Users can click the Add button in the table to add rows. The data that users enter must meet the validation criteria for each cell, such as minimum text length, data format, and so forth. For each cell, the type of element that you specify determines this validation criteria.

■   Users *cannot* add, delete, move, or edit rows that you specify when you create the form. All rows inserted by a user are marked with red dots in their cells.

■   When users are adding or updating rows, JavaScript functions run *only* on the row being added or updated.

■   User-created rows always appear at the *top* of the table, *before* all rows populated through a report data object, an API plug-in, or static data. Moreover, user-created rows always appear in the reverse order that they were added: The first rows in the table are the last rows added.

■   Sorting does not apply to user-created rows. However, sorting does apply to the rows populated through an API plug-in or static data.

Auto Number

Specifies whether the first column of each row contains the row number.

If the value is true, then the first column of each row displays the row number.

The default is true.

Height

Specifies the height of the table, in pixels.

The default is 125.

Maximum Selected Rows

Specifies the maximum of rows that users can select in the table while completing the form in a request.

If the number of selections exceeds this value, an error occurs.

If this value is not specified, then the number of rows that users can select is unlimited.

Minimum Selected Rows

Specifies the minimum of rows that users can select in the table while completing the form in a request.

If this value is not specified, then the user is not required to select any rows.

Page Size

Specifies the number of rows to display on each page of the table.

The same text applies as the Page Size attribute for a select box (see page 308).

A blank (empty) value effectively disables pagination.

The default is 50.

Mode

Specifies whether the table functions in *selection mode* or *data mode* during the request lifecycle, as follows:

■   In selection mode, the table automatically includes a check box in the first column, so that users can select one or more rows from the table. The user-selected rows appear in the request throughout the request lifecycle.

■   In data mode, the table does *not* include the check box. Therefore, users cannot select rows from the table.

In *both* modes, if the value of the Allow Inserts attribute (described earlier in this topic) is True users *can* do the following: Configure the request by adding rows to the table and entering custom values. The catalog system automatically saves any rows that the user adds and displays them throughout the request lifecycle. Users can remove each row they added by clicking the row and selecting the Remove button.

Selection mode is useful when you want users to select a set of related options for the request by selecting a row on the table. For example, you can specify a table like the following:

Check Box	Disk Space, GB	Memory, GB	Processor Speed, GHz	Operating System
_	700	4	2	Windows XP
_	900	6	2.4	Windows 7
_	1000	8	3	Windows 2008 Server
_	1400	8	4	Windows 7 Server

In this example, each row contains one set of configuration options for a computer reservation. Users cannot select each option individually, but they can select a set by selecting a row.

Attributes for the Selected Row Only

The following table attributes apply to the selected row only, not the entire table:

Column Widths

Specifies a list of comma-separated values for the *widths* of each column in the table, from left to right.

For example, if the table contains three columns, then the following is true:

■   The first value applies to the first (left) column

■   The second value applies to the second (middle) column

■   The third value applies to the third (right) column

Note: This attribute *applies* to the entire table, but *appears* only in the first row.

Column

Applies to static tables (see page 288) *only, after* you add at least one element to the table. Each element that you add creates a column. Does not apply to dynamic tables (see page 290).

Specifies a list of comma-separated values for each column in the table, from left to right. When users open the form in a request, these values appear in the columns as prefilled entries.

For example, if the table contains three columns, then the following is true:

- The first value applies to the first (left) column

- The second value applies to the second (middle) column

- The third value applies to the third (right) column

Note: This attribute *applies* to the entire table, but *appears* only in the first row.

Attributes for Text Fields and Text Areas Only

The following HTML attributes (see page 303) apply to text fields and text areas *only*. Text fields and text areas are basic elements of a form (see page 281).

password

Specifies an attribute for text fields. Can have a value of true or false. When the value is set to true, this component becomes a password component.

Height

Specifies the number of rows in the text area. For example, if you specify 3, the text area contains three rows.

Maximum Length

Specifies the maximum number of characters that the user can enter in the field.

Pattern

In the value for this attribute, you specify regular expressions to validate numeric and address data (see page 361). Such data includes credit card numbers, social security numbers, email addresses, IP addresses, and telephone numbers.

Pattern Message

Applies only when *pattern* (the previous attribute) is specified.

You can specify an error message to appear when the user input violates the pattern attribute. You can optionally localize (see page 375) this error message.

JavaScript Attributes for Elements

You can use JavaScript attributes to invoke JavaScript Functions (see page 341) while the user is completing the form in a request. Examples include onChange, onKeyUp, onBlur, and so forth. JavaScript functions include predefined functions for validating user input into fields and any custom functions that you have written. All JavaScript attributes are named with the prefix *on*, such as onClick.

You can use the following JavaScript attributes in elements on forms. Not all attributes apply to all elements, however. Click an element to see which attributes apply to it.

Important! JavaScript attributes *must* have JavaScript functions but not JavaScript expressions as values. Conversely, HTML attributes (see page 303) can have JavaScript expressions but *not* JavaScript functions as values. Thus, JavaScript expressions apply to HTML attributes *only*, and JavaScript functions apply to JavaScript attributes only.

JavaScript functions specified in JavaScript attributes are validated when the user performs the action specified by the attribute, such as click or double-click. If the field does not validate, it is highlighted in red and an error message appears explaining why the validation failed.

Note: Verify that the error messages (if any) returned by each JavaScript function are localized for the users of the form.

onFocus

Specifies the JavaScript function to run when the element comes into focus.

onBlur

Specifies the JavaScript function to run when the element loses the focus.

onChange

Specifies the JavaScript function to run when the element value is changed.

onClick

Specifies the JavaScript function to run when the component is clicked with the left mouse button.

onMouseDown

Specifies the JavaScript function to run when a mouse button is pressed.

onMouseUp

Specifies the JavaScript function to run when a mouse button is released.

onMouseOver

Specifies the JavaScript function to run when the component is moused over.

onMouseMove

Specifies the JavaScript function to run when the mouse passes over the component.

onMouseOut

Specifies the JavaScript function to run when the mouse was moved away from the component.

onKeyPress

Specifies the JavaScript function to run when a key is pressed and released.

onKeyDown

Specifies the JavaScript function to run when a key is pressed.

onKeyUp

Specifies the JavaScript function to run when a key is released.

onValidate

Specifies the JavaScript function to run when the field is validated. The field is validated whenever the user navigates away from a field or when the user submits the form.

If you specify a custom JavaScript function for the onValidate attribute, code that function to do the following:

- Specify an attribute named _val; this attribute is required to pass the current value of the field.

- If no validation error occurs, return null.

- If a validation error occurs, return a string with a helpful error message explaining the problem and solution. Localize error messages for each browser locale applicable to your users.

onLookup

Applies to lookup fields *only*. When the user clicks the magnifying glass for the lookup field, the JavaScript function specified by this attribute runs. You can use a lookup field to

How to Perform Automated Tasks for Fields

As an administrator, you can optionally use JavaScript expressions and JavaScript functions to perform automated tasks in the fields of forms created using the Form Designer. The possible means of using JavaScript expressions and JavaScript functions are nearly endless, using the rules and guidelines explained in this topic. At runtime, when the user fills out the form while completing a request, the JavaScript expression or function runs and performs the specified task.

Important! As a pre-requisite for using JavaScript expressions and functions, you must have a working knowledge of HTML, CA Service Catalog administration, and JavaScript.

Follow this process to specify automated tasks in fields, using one or more of the following: JavaScript expressions, JavaScript functions, and report data objects.

1. Create the form (see page 296), if you have not already done so.

2. For the applicable field or fields, decide the automated task that you want to perform. For example, you may want to pre-populate fields, as described earlier in this topic. Alternatively, you may want to hide or disable forms or fields under certain specific conditions. Common examples are covered later in this topic.

3. Verify that you can represent the field or fields as one or more of the elements (see page 281) available for creating forms, such as text fields, check boxes, select boxes, or radio groups.

4. Determine whether to achieve the purpose or goal you intend for the field by one or both of the following methods:

   ■   Specifying a JavaScript *expression* for the value of an *HTML* attribute of the element

   ■   Specifying a custom or pre-defined JavaScript *function* for the value of a *JavaScript* attribute of the element

   Both methods are summarized in the next steps. Before finalizing your decision, read all documentation about both methods. Doing so helps you verify that the end results achieve your goal for the affected fields.

5. Consider specifying a JavaScript expression for the value of an HTML attribute of the element. The JavaScript expression runs only once, as soon as the form appears.

   Examples include the following:

   ■   Pre-populating user-related data in one or more fields on a form, based on the user completing the form

   ■   Disabling a field based on the request status, on the business unit or role of the user completing the form, or on other criteria

   ■   Hiding a field based on the request status, on the business unit or role of the user completing the form, or on other criteria

JavaScript expressions retrieve the values of object properties from the CA Service Catalog database at runtime. These properties are grouped into the following categories: user, business unit, service, service option groups, and request.

You can specify JavaScript expressions in fields (see page 325) for the values of certain HTML attributes.

Note: To hide an *entire* form, based on role, business unit, request status, or other criteria, you do not specify JavaScript expressions for any fields but rather for the entire form when you attach the form to a service option group (see page 366).

6. Consider specifying a pre-defined JavaScript function (see page 342) or a custom JavaScript function for the value of a JavaScript attribute of the element. This method is best suited for dynamic events that occur when the user performs an action while completing the form. In other words, the JavaScript function can run multiple times when a user checks a box, enters a value, or performs some other action. Thus, JavaScript functions are most useful when you want to create dynamic relationships between fields.

   Examples include the following:

   ■ Retrieving the values of one or more object properties from the CA Service Catalog database, the CA MDB, or other data source. This option especially useful when the value *cannot* be retrieved with a JavaScript expression. Consider all options to populate and pre-populate fields automatically (see page 319).

   ■ Validating user input (see page 359), specifically validating the format of numeric or address data entered by the users, including social security numbers, credit card numbers, telephone numbers, email addresses, and IP addresses.

   ■ Enabling or displaying a list of options or fields under a check box when a user clicks the box to check it

   ■ Disabling or hiding a list of options or fields under a check box when a user clicks the box to un-check it.

7. For more information, see How to Use JavaScript Functions in Fields (see page 341). Review and follow the instructions in the cross-referenced topics in the previous steps to create your JavaScript expression, JavaScript function, or both. Those sections explain how to accomplish specific tasks in detail.

8. Test the JavaScript expressions and JavaScript functions to verify that they accomplish your goals for the fields on the form.

9. As a best practice, verify that you test the expression in a form used with a service in a test environment before you use the form and the service in a production environment.

Options to Populate and Pre-populate Fields Automatically

The Form Designer provides several options for populating and pre-populating fields automatically. *Populating* fields means that the fields are filled with the results of a database query *based on* user input. The affected fields are empty when the user opens the form. When the user enters the requested input, the database query is run and the results are used to populate the fields.

Conversely, *pre-populating* fields means that the fields are filled with the results of a database query *without* any user input. As soon as the user opens the form, the data source query is run and the results are used to populate the fields. Thus, the affected fields are populated when the user opens the form.

Review the options and choose the one that best suits your needs. Use the pre-populating options when no user input is required for the requested information to be retrieved by the database query. Use the populating options when the user's input is critical for the requested information to be retrieved by the database query.

The options for populating fields follow:

■ Use the pre-defined JavaScript function (see page 342) named ca_fdDoFieldLookup with a lookup field to populate fields based on user input to a report data object (see page 351).

■ Use ca_fdDoFieldLookup with a lookup field, and use other pre-defined JavaScript functions like ca_fdSetTextFieldValue to set the values of elements in your form with a custom JavaScript function.

The options for pre-populating fields follow:

■ Pre-populate a combo box based on a report data object (see page 320).

■ Use ca_reportQuery and ca_fdSetTextFieldValue to pre-populate fields based on a report data object and JavaScript functions (see page 356).

■ Pre-populate fields based on JavaScript expressions (see page 335).

How to Pre-Populate a Combo Box Based on a Report Data Object

CA Service Catalog provides several options for pre-populating fields on a form based on a report data object (data object), without depending on any data entered by the user and without using JavaScript expressions or JavaScript functions. This topic explains one of these options. Specifically, this topic explains how to pre-populate a combo box using a data object. A combo box is a combination text box and list. On a form, a combo box appears as a single line item; the user clicks its arrow to open the entire list. You configure the combo box to specify whether the user can select only one option or multiple options.

On the Form Designer, you add and configure the select group element (see page 281) to function as a combo box. To pre-populate the combo box with the results of the data object, you use the HTML Attribute (see page 303) named reportobjid. When the user opens the form, the data object runs and return the results in the combo box. The user then selects one or more of the options in the combo box while completing the form.

This option is useful when you want to provide the user with multiple valid choices from your data source but do not want to enable the user to specify a custom selection. This option thus helps enforce the standardization and validity of users' selections while still typically providing users with multiple options from which to choose.

To pre-populate a combo box using a data object, follow this process:

1. In the Report Builder, create a data object (see page 99) that queries the data source (such as the MDB) for the data you want. The query must specify *two* fields from the data source, as follows:

    ■   The first field returns the list of all possible values and populates the combo box with this list.

    ■   The second field returns the label for the combo box.

    When the form appears, the report data object runs and populates the combo box with the resulting data.

    Record the name of the data object for later reference.

    Consider the following sample query for a data object for the MDB. This example returns the list of available service option groups (rate plans).

    SELECT rate_plan_id,rate_plan_name FROM usm_rate_plan

This query does the following:

- "SELECT rate_plan_id,rate_plan_name" specifies which values to return:

  – rate_plan_id returns the list of all service option groups. Each plan becomes an option in the combo box; for example, Budget, Standard, Deluxe, and so forth.

  – rate_plan_name returns the label for the combo box; for example, Service Option Groups.

- "FROM usm_rate_plan" specifies that the data source is the usm_rate_plan table in the MDB.

2. On the form, add a select box; it is one of the elements of a form (see page 281).

    a. For the _id attribute, specify a meaningful name, and save the form.

    b. For the value of the HTML Attribute (see page 303) named reportobjid, specify the id of the data object, and save the form.

    Note: When you use a data object to populate a combo box, do not add any options to the select box, because they are ignored (not used) when the user opens the form. Any options for the select box are ignored, only the data object "matters."

    c. Verify that the *multiple* attribute of the select box meets your requirements:

    If you specify False, users can select only one option in the combo box.

    If you specify True, users can select multiple options in the combo box.

    d. For the title attribute (tool tip text), optionally specify instructional text, such as "Click the arrow and scroll to select a value." Save the form.

    Note: You can optionally localize (see page 375) this tool tip text if users from different locales use this form.

How to Use User Input to Pre-Populate a Select Box

You can use input provided by the user in one or more fields in the form to determine the values in a select box. This technique is useful when you want to provide the user with two sets of multiple valid choices from your data source but do not want to enable the user to specify a custom selection. This option thus helps enforce the standardization and validity of users' selections while still typically providing users with multiple options from which to choose.

A common application of this technique occurs on a form with select boxes for both country and state. Using the reportobjid and reportobjvars attributes, you can configure the state select box to display only the states for the country selected by the user.

To use this technique of pre-populating a select box based on a user's selection in an earlier select box, follow this process. This process uses the country-state scenario as a running example.

1. In the Report Builder, create a data object (see page 99) that queries the data source (such as the MDB) for the data you want in the first select box. The query must return the list of all possible values and populate the combo box with this list. This step is part of the process of pre-populating a combo box based on a report data object (see page 320).

    In our running example, create a report data object to retrieve the list of countries from the database.

    When the form appears, the report data object runs and populates the combo box with the resulting data.

    Record the ID of the data object for later reference.

    Consider the following sample query for a data object for the MDB. This example returns the list of available countries for the service or service option to which the form is attached

    SELECT country_id,country_name from my_country_table

    This query does the following:

    ■   "SELECT " specifies which values to return. In this example, it returns the ID and the name of the country in the database.

    ■   "FROM my_country_table" specifies the name of the database table.

2. On the form, add the first select box; it is one of the elements of a form (see page 281).

    a. For the _id attribute, specify a meaningful value, and save the form. Record the value for later reference when you specify the reportobjvars attribute.

    b. For the value of the HTML Attribute (see page 303) named reportobjid, specify the id of the data object, and save the form.

    Note: When you use a data object to populate a combo box, do not add any options to the select box, because they are ignored (not used) when the user opens the form. Any options for the select box are ignored, only the data object "matters."

    c. For the *multiple* attribute of the select box, specify False so that users can select only one option in the combo box. Save the form.

    d. For the title attribute (tool tip text), optionally specify instructional text, such as "Click the arrow and scroll to select a country." Save the form.

    e. Optionally rename the default display text from the select group from "Select" to a more meaningful name, by selecting the element in the Component Tree and clicking the Rename icon at the top of the tree.

    Note: You can optionally localize (see page 375) the tool tip text or the name of the select box if users from different locales use this form.

    f. Leave the value attribute empty. This value is populated with the first result returned by the query.

3. Create a second report data object to retrieve a list of values from the database, based on the user's selection in the first select box. In this report data object, specify some report variables in the query that are to be filled using the input provided by user in other fields in the form

    In our running example, create a second report data object to retrieve the states from the database, based on the country that the user selected. Follow the guidelines in Step 1 to create the second report data object.

4. On the form, add the second select box. This box is to be pre-populated, based on the user's selection for the first select box. Follow the guidelines in Step 2 to create the second select box, but account for the following considerations:

    ■ The second query contains a report variable and is similar to the following:

    select state_id,state_name from my_state_table where country_id=%selected_country%

    ■ The running example in this topic uses only one report variable (%selected_country%); however, your query may contain multiple report variables.

    ■ Record the names of all report variables for use when you specify the reportobjvars attribute for the second select box.

5. In the second select box, specify the reportobjvars attribute using the following format: $({'reportvar':*value*}).

    A valid *value* is one of the following:

    ■   A constant, such as 12, as in $({'reportvar':12})

    ■   A string, such as abc, as in $({'reportvar':'abc'})

        Note: Enclose strings in single quotation marks, as shown. Within a quoted string, if needed, use a backslash (****) as an escape character to specify a literal single quote or apostrophe. For example: $({'what\\'s the status?'})

    ■   A JavaScript expression, such as the user's last name, as in $({'reportvar':_.user.lastName})

    ■   A JavaScript function of the following form: $({'reportvar':foo()})

    ■   Combinations of the previous values, delimited with commas, using the format shown in the following example:

        $({'reportvar':_.user.lastName,'reportvar1':'abc','reportvar2':12,'reportvar3':foo()})

In our running example, create a second select box to be populated with the states from the database, based on the country that the user selected. For the reportobjvars attribute of the state select box, specify the pre-defined JavaScript function named ca_fdGetSelectedOptionValues, as follows:

$({'selected_country':ca_fdGetSelectedOptionValues ('<form _id> ','country')[0]})

form _id

Specifies the value of the _id attribute of the form containing the first select box. You reference this form in Step 2.

country

Specifies the value of the _id attribute of the first select box. You create and record this value in Step 2.

6. In the first select box, set the onchange attribute to retrieve the data for the second select box as soon as the user makes a selection in the first select box. For the onchange attribute, specify the pre-defined JavaScript function named ca_fdFetchSelectData, using the following format:

    ca_fdFetchSelectData('<form _id>','<*field _id*>');

    form _id

    > Specifies the value of the _id attribute of the form containing the second select box. You reference this form in Step 4.

    field_id

    > Specifies the value of _id attribute of the form containing the second select box. You create and record this value in Step 4.

    > In our running example, set the onchange attribute for the country select box to the following:

    > ca_fdFetchSelectData('<form _id>','<state field _id>');

7. Test the form to verify that it works as you intended.

As a best practice, test this form with a service in a test environment before you use them in a production environment.

How to Use JavaScript Expressions in Fields

There are many possible applications of using JavaScript expressions to perform automated tasks in fields on forms. One of the most common and most useful applications is to pre-populate the value attribute of a field with the runtime value of one of many objects and properties from the CA Service Catalog database. To pre-populate a field for a user while the user is completing a form, you may specify a simple JavaScript expression for the value attribute of the field, using the following format: $(_.*object*.*property*), such as $(_.user.lastName).

To specify JavaScript expressions in forms, do the following:

1. If you have not already done so, review the information in <u>How to Perform Automated Tasks for Fields</u> (see page 317) to help decide whether you want to use a JavaScript expression or a JavaScript function to perform an automated task for one or fields on a form.

2. Decide the fields on the form for which you want to specify a JavaScript expression, and the task that you want the JavaScript expression to perform. For example, you may want to pre-populate fields, as described earlier in this topic. Alternatively, you may want to hide or disable forms or fields under certain specific conditions. Common examples are described later in this topic.

3. Verify that you can represent the field as one of the [elements](#) (see page 281) available for creating forms, such as text field.

4. Verify that the purpose or goal you intend for the field is represented in one of the following [HTML attributes](#) (see page 303). You can specify JavaScript expression for the values of these HTML attributes *only*.

   - Value   - Value is the only attribute for which you can use JavaScript expressions to pre-populate values in fields. The value attribute is most frequently used for this purpose, especially for pre-populating user-related data such as address, phone numbers, business unit, and other personnel-related data

   - Disabled - You can disable a field, according to the user's role or business unit, the request status, or other criteria that you specify.

   - Hidden - You can hide a field, according to the user's role or business unit, the request status, or other criteria that you specify.

   - Checked - Check boxes and radio buttons support the checked attribute.

     You can select a check box or a radio button, according to the user's role or business unit, the request status, or other criteria that you specify.

   - Required - You can set a field as required, according to the user's role or business unit, the request status, or other criteria that you specify.

5. Keep in mind these rules when coding JavaScript expressions for use in the HTML attributes:

   - For the disabled, hidden, checked, and required attributes, the JavaScript expression must return a value of true or false.

   - If the returned value is any value except true, CA Service Catalog automatically uses a value of false. The value attribute is the only attribute that is used for pre-populating fields.

6. Review and follow the instructions for performing any of the following tasks or use them as a model for a similar task, if applicable:

   - [Pre-populate fields](#) (see page 335)

   - [Hide or disable a field](#) (see page 336) based on the request status, the business unit or role of the user completing the form, or on other criteria

   - [Select options for fields by default](#) (see page 339) based on the request status, the business unit or role of the user completing the form, or on other criteria; you may select a check box or an option in a radio group by default, based on such criteria

7. Review the objects and properties that you can specify in JavaScript expressions (see page 327) and determine which ones you need. These objects and properties are retrieved from the CA Service Catalog database at runtime, as noted earlier, and are grouped according the following categories: user, business unit, service, service option group, and request.

   If the objects and properties that you need cannot be retrieved using a JavaScript expression, consider using the following pre-defined JavaScript function (see page 342) to retrieve them: ca_reportQuery(reportId, variables, onSuccess, onFailure).

8. Using all applicable information from the previous steps, construct and test the JavaScript expression to accomplish your goals for the fields on the form.

9. As a best practice, verify that you test the expression in a form and a service in a test environment before you use the form and service in a production environment.

Objects and Properties that You Can Specify in JavaScript Expressions

You specify objects and properties in JavaScript expressions. You can use these expressions for the runtime values of the *value* attribute for the elements that you use as fields on forms. Typically, you specify these expressions in the following format: $(_.*object*.*property*).

For example, to capture the runtime value of the logged in user completing the form in a request, specify $(_.user.firstName). You can also concatenate strings, as explained later in this topic.

These variables apply for Form Designer forms *only*! They are similar but are not the same as the objects and properties defined for CA Workflow forms variables (see page 202). If you are using both Form Designer forms and CA Workflow forms, use the correct objects and properties for each type of form.

Forms

The form object contains the following property that you can use in JavaScript expressions or as a first parameter in JavaScript functions (see page 342): ca_fd.formId.

formId refers to the *active* form. The form is in active when a field in the form triggers an event. The following are a few sample actions that activate forms:

- The user opens the form while working on a request, which activates the JavaScript attribute named onLoad.

- The user clicks a label or image, which activates the JavaScript attribute (see page 315) named onClick.

- The user submits the form, which activates the JavaScript attribute named onSubmit.

User

The user object contains an array of user properties that you can access using _.user.

Each user object has the following properties, listed in related groups:

- id, uuid, status (0 = active, 1 = inactive)

- firstName, lastName, middleName, commonName, alias, title

- groups

  The _.user.groups property specifies the name of a user group in CA EEM. You can use this property to manipulate a field based on whether the user belongs to a specific CA EEM group. For more details, see the Example section.

  The _.user.groups property contains an array with all the CA EEM groups to which the user belongs.

- manager, delegate, description

- phone (an array: phone[0] = primary and phone[1] = secondary)

- mobile, fax, pager, email

- timezone, localeLanguage, localeCountry, defaultRole, defaultDomain, location.uuid, location.name, location.city, location.state, location.country, location.postalCode, location.phone, location.fax, location.description, location.address[0-5]

- roles.<domain>

Examples

You can use the _.user.groups property to hide a field from members of a CA EEMgroup, for example, the group named developers. To do so, set the value of the HTML attribute (see page 303) named Hidden to the following:

_.user.groups.indexOf("developers") >= 0

In contrast, you can also make a field visible to users who are *not* members of an CA EEM group. To do so, set the value of the Hidden attribute to the following:

_.user.groups.indexOf('developers') < 0

Business Unit

The business unit object contains an array of business unit properties that you can access using _.bu.

Each business unit object has the following properties, listed in related groups:

- id, name

- type(business unit type where SP=service provider, ST=can have child business units, TE=cannot have child business units)

- singleAccountMode (true or false)

- status (0=inactive, 1=active)

- openedDate, description, timezone, federalTaxId, stateTaxId, taxRegion, currency, dateFormat, parent (parent bu id)

- email, website, primaryContact (contact userid), location.uuid, location.name, location.city, location.state, location.country, location.postalCode, location.phone, location.fax, location.description, location.address[0-5]

- data1, data2, data3, data4, and data5

Request

The request object contains an array of request properties that you can access using _.request.

Each request object has the following properties, listed in related groups:

- id, name, requestedFor, requstedForAccountId, requestedBy, requestedByAccountId

- description, priority, status

- dateCreated, completionDate, dateRequired, and lastModified

- newDateRequired, newName, and newPriority

  These properties reflect the new value of an object as soon as the user updates the object during the request lifecycle. (The user is typically a request manager or the requestor.)

  The user can update the object, for example, while approving, rejecting, or pushing through the pending action for the item associated with the property. The *new* value takes effect *immediately* when the user updates the property; saving is *not* required.

  When creating or editing forms, you can use these properties at your discretion. For example, you can use these properties in a custom JavaScript function for validation that you use in the JavaScript attribute (see page 315) named OnSubmit. For example, if a user enters an invalid date such as a holiday, your custom JavaScript function can display text explaining the valid dates.

Service

The service object contains an array of services that you can access using _.service.

Each service object has the following properties, listed in related groups:

- id, bu, name, description

- status(0=deleted, 1=available, 2=unavailable, 3=created, 4=cancelled, 5=total)

- website, code, version, dateAvailable, dateUnavailable, dateCreated, dateCancelled

Service Option Groups

The service option groups object contains an array of service options that you can be access using _.sog.name.

Each service option group has the following properties, listed in related groups:

- id, bu, name, description

- status(0=deleted, 1=available, 2=unavailable, 3=created, 4=cancelled, 5=total)

- code

- dateAvailable, dateUnavailable, dateCreated, dateCancelled

Service Options

The service option object applies *only* to the predefined JavaScript function (see page 342) named _.serviceoption.status( ).

The _.serviceoption.status( ) function contains a single property, which is named newStatus.

Note: This function does *not* take any parameters.

newStatus--

Specifies the *new* value for the status when a user action updates the status of the service option. This property takes the new value for the status as soon as the user does one of the following:

■ Selects a new status value from the drop-down list in the Action column while managing a requests pending action (see page 604). A request manager typically performs this action. The value of newStatus changes dynamically as soon as the user selects the new status of the service option.

■ Opens a page or dialog that changes the status to a fixed value when the user clicks OK. For example, when a user creates a request, the value of newStatus changes dynamically as soon as the user creates the request. In this example, the value of newStatus immediately becomes Submitted.

In both cases, the value of _.serviceoption.newStatus() changes immediately. In contrast, the value of _.serviceoption.status( ) remains unchanged until the page is submitted and refreshed.

Note: JavaScript functions related to status changes do *not* change their behavior if they depend on the user completing the page or dialog. (Typically, this action is clicking OK). For example, the JavaScript function named onSubmit typically references status changes. The onSubmit function runs *only* when the user completes the page or dialog, *not* when the value of newStatus changes.

Examples

You can use the newStatus property to help your implementation process reservations efficiently. For example, suppose a user submits a main request for a computer and additional requests for related options. These options include extra memory, an upgraded keyboard, and so forth. You can create a custom JavaScript function so that if the main request is rejected, the onSubmit function runs for all optional requests. Examples of optional requests include accessories, extra memory, and so forth. For example, the following expression evaluates to True when the original status is 400 and the newStatus is changed to 800:

```
if (_.serviceoption.status() == 400 && _.serviceoption.newStatus == 800)...
```

Operators

The operators most commonly used for form design are described here. For complete details about standard operators, see the JavaScript standards reference that your organization uses, for example, www.developers.sun.com, or www.javascript.com.

You can specify all standard operators in your JavaScript expression *except* assignment operators.

The assignment operators are as follows: =, +=, -=, *=, /= .

For example, the folowing expression is invalid because it uses the = assignment operator:

$(var x = 1+2)

Required Return Values

Specify JavaScript expressions for the disabled, checked, and hidden attributes that return one of the following values:

- true or "true"
- false or "false"

If the expression returns any other value, CA Service Catalog replaces it with a value of false. Therefore, if you specify $(_.user.firstName) for the disabled attribute of a text field, the field is not disabled unless "true" is the first name of the user.

Concatenation Operators

You can optionally concatenate two strings together, using the + operator,

For example, $('Hello ' + _.user.firstName + ' ' + _.user.lastName) returns the following text: Hello *first-name last-name.* "Hello John Doe" and "Hello Jane Smith" are samples.

Comparison Operators

You can optionally use the following comparison operators:

- == is equal to

- === is exactly equal to (value and type)

- != is not equal

- > is greater than

- < is less than

- >= is greater than or equal to

- <= is less than or equal to

For examples used in the Form Designer, see the topics for the following:

- Hiding or disabling a field based on request status, business unit, role, or other criteria (see page 336)

- Selecting or clearing fields by default based on request status, role, business unit, or other criteria (see page 339)

Logical Operators

You can optionally use the following logical operators:

- and (&&)

- or (||)

- not (!)

For examples used in the Form Designer, see the topics for the following:

- Hiding or disabling a field based on request status, business unit, role, or other criteria (see page 336)

- Selecting or clearing fields by default based on request status, role, business unit, or other criteria (see page 339)

How to Pre-Populate Fields Based on JavaScript Expressions

When you use an element (see page 281) that includes the *value* attribute as a field on a form, you can use a JavaScript expression to pre-populate the element's value attribute with the runtime value of one of many objects and properties from the CA Service Catalog database.

For example, in the text field for First Name on your form, you can specify the JavaScript expression $(_.user.firstName) to pre-populate the field with the first name of the logged in user ID who is creating the request and completing the form. Similarly, in the text field for Last Name, you can specify the JavaScript expression $(_.user.lastName) to pre-populate the field with the user's last name.

To pre-populate fields on forms using JavaScript expressions, follow this process:

1. Design and create the form (see page 296).

2. Verify that the best way to perform the automated task (see page 317) in the field or fields of interest is to use a JavaScript expression.

3. Review the guidelines for using JavaScript expressions in fields (see page 325) and verify that you want to pre-populate the value attribute of an element in the field.

4. Verify that the data that you want to pre-populate is one of the objects and properties that you can specify in JavaScript expressions (see page 327). These objects and properties are related to the logged in user's personnel data, to one or more business units, or to service, service options, status, or other data related to the request that contains the form.

5. Specify the JavaScript expression in the value attribute of the element for the field. When specifying the expression, follow all syntax rules in objects and properties that you can specify in JavaScript expressions (see page 327), especially the rules related to using operators to link properties. The following are some valid expressions that you may find useful as-is or as models:

   ■ User's first name: $(_.user.firstName)

   ■ User's last name: $(_.user.lastName)

   ■ User's first name and last name, concatenated in a single field: $(_.user.firstName + ' ' + _.user.lastName)

   You can use this example to pre-populate a text field at the end of the form, such as a thank you message.

   ■ User's city: $(_.user.location.city)

   ■ User's state: $(_.user.location.state)

   ■ User's physical address data, concatenated in a single field: $(_.user.location.address[0] + ' ' + _.user.location.address[1] + ' ' + _.user.location.city + ' ' + _.user.location.state)

   This example returns the street address, city, and state of the user, for an address in the United States.

- User's role: $(_.user.roles[domainId])

- User's business unit, with parent: $(_.bu.id)

- User's business unit, without parent: $(_.bu.id.parent)

- User ID for whom the request is made: $(_.request.requestedFor)

- User ID who made the request: $(_.request.requestedBy)

- Name of the service associated to the form: $(_.service.name)

- Status of the service associated to the form: $(_.service.status)

- Name of service option group in the service associated to the form: $(_.sog.name)

- Description of service option group in the service associated to the form: $(_.sog.name.description)

- Status of service option group in the service associated to the form: $(_.sog.name.status)

6. Test the JavaScript expression to verify that it accomplishes your goals for pre-populating the field on the form.

7. As a best practice, verify that you test the expression in a form used with a service in a test environment before you use the form and the service in a production environment.

How to Hide or Disable a Field Based on Request Status, Business Unit, Role or Other Criteria

You may decide to use JavaScript expressions to hide or disable a field based on the request status, on the business unit or role of the user completing the form, or on other criteria.

As an administrator, you typically disable fields that you want users to view but *not* update. Examples may include options for a service that the user's manager selected or that the request manager selected based on available inventory.

In contrast, as an administrator, you can hide fields when you either do not want users to be aware of them at all, for some reason, or when the fields add distracting information that the user does not need. Examples include cost data, inventory options available to certain roles only or certain business units only, or data that does affect the end user, such as the estimated administrative cost of fulfilling the service.

To hide a field, specify the JavaScript expression that specifies your criteria in the hidden attribute of the field element. Similarly, to disable a field, specify the JavaScript expression that specifies your criteria in the disabled attribute of the field element.

Important! Hiding or disabling a *field* is not completely secure. While the field and its value are not displayed on the CA Service Catalog page, the data does still exist on the form's HTML page and can be accessed by viewing the source text of the browser. Therefore, to restrict sensitive data, create two versions of the form and hide the form that contains the sensitive data from users who you do not want to see it. For more information, see How to Hide, Enable, or Disable an Entire Form Under Specified Conditions (see page 340).

To disable or enable a field based on your criteria, follow this process:

1. Design and create the form (see page 296).

2. On the Form Designer, view the field that you want to enable or disable and verify that it includes the HTML attribute (see page 303) named disabled.

3. Determine the exact criteria that you want to use to disable or enable the field. The criteria may be based on the request status, on the business unit or role of the user completing the form, or some other criteria.

   If the disabled attribute is set to true, users can view the field but cannot edit it. Conversely, if the disabled attribute is set to false, users can both view and edit the field.

4. Verify that the data that you want to use to determine whether to enable or disable the field is one of the objects and properties that you can specify in JavaScript expressions (see page 327). These objects and properties are related to the logged in user's personnel data, to one or more business units, or to service, service options, status, or other data related to the request that contains the form.

5. Review the guidelines for using JavaScript expressions in fields (see page 325).

6. Specify the JavaScript expression in the value attribute of the element for the field. When specifying the expression, follow all syntax rules in objects and properties that you can specify in JavaScript expressions (see page 327), especially the ones related to using operators to link properties. The following are some valid expressions that you may find useful as-is or as models:

   ■ To hide or disable a field if the user has the enduser role, set the hidden or disabled attribute of that field to the following JavaScript expression: $(_.user.roles[_bu.id]=='enduser').

      This expression returns a value of true and therefore hides or disables the field *only* if the user filling out the form has an enduser role in the current business unit.

You may use this setting or a similar one to restrict users from either viewing or editing the text field named Memory in a form for a request for a new laptop computer. Conversely, this example displays or enables the field for all other roles in the business unit. Thus, when used with the disabled attribute, this example permits the request manager to edit the field, based on the available inventory.

To display or enable a field for only request managers, hide or disable it for all other roles, by setting the hidden or disabled attribute of that field to $(_.user.roles[_.bu.id] != 'requestmanager').

- To hide or disable a field in a form shared by all business units *only* if the current business unit is PBJ222, set the hidden or disabled attribute of that field to $(_bu.id=='PBJ222').

- To hide or disable a field in a form shared by all business units *only* if the parent of the current business unit is PBJ222, set the hidden or disabled attribute of that field to $(_bu.id.parent=='PBJ222').

- To hide or disable a field if the request was created after a certain date, set the hidden or disabled attribute of that field to $(new Date(_.request.dateCreated) < 'date').

  For example, to hide or disable a field if the request was created after January 15, 2010, specify $(new Date(_.request.dateCreated) < new Date().setFullYear(2010,0,15)).

- To hide or disable a field if the request status is Approval Done or higher, set the hidden or disabled attribute of that field to $(_.request.status=>600).

7. Test the JavaScript expression to verify that it accomplishes your goals for enabling or disabling the field on the form.

8. As a best practice, verify that you test the expression in a form used with a service in a test environment before you use the form and the service in a production environment.

How to Select Options for Fields by Default Based on Request Status, Role, Business Unit, or Other Criteria

As an administrator, you may decide to use JavaScript expressions to select a check box by default or select an option for a radio group by default, based on the request status, on the business unit or role of the user completing the form, or on other criteria. Conversely, you may want to clear certain check boxes or radio group options by default according to such criteria. An example is selecting check boxes related to publicly known managerial duties by default for managers only, while clearing them for other users. Another example is selecting software-related radio group options for the Software business unit, while clearing them for other users.

Remember, of course, that selecting or clearing the check box or radio group option by default does not prevent the user from changing the default you specified before submitting the request that contains the form. Therefore, before using this feature, verify that permitting users to change the default is appropriate. Otherwise, consider redesigning the form to either hide or disable the field (see page 336) the options that you want to prohibit users from changing, according to the criteria specified in your JavaScript expression.

To select a check box or radio group option by default based on your criteria, follow this process:

1. Design and create the form (see page 296).

2. On the Form Designer, view the check box or radio group option that you want to select by default and verify that it includes the HTML attribute (see page 303) named checked.

3. Determine the exact criteria that you want to use to select the check box or radio group option by default. The criteria may be based on the request status, on the business unit or role of the user completing the form, or some other criteria.

4. Verify that the data that you want to use to make the selection is one of the objects and properties that you can specify in JavaScript expressions (see page 327). These objects and properties are related to the logged in user's personnel data, to one or more business units, or to service, service options, status, or other data related to the request that contains the form.

5. Review the guidelines for using JavaScript expressions in fields (see page 325).

6. To select a check box or select an option for a radio group by default, specify the JavaScript expression that specifies your criteria in the checked attribute of the field element. When the user opens the form, the expression runs as follows:

   ■ If the criteria *are* satisfied, then value returned is true and therefore the check box or radio group option is selected by default.

   ■ If the criteria are *not* satisfied, then value returned is false and therefore the check box or radio group option is *not* selected by default.

When specifying the expression, follow all syntax rules in objects and properties that you can specify in JavaScript expressions (see page 327), especially those related to using operators to link properties. The following are some valid expressions that you may find useful as-is or as models. In these examples, the term *option* refers to both a check box and a radio group option.

■ To select an option by default if the user has the enduser role, set the checked attribute of that field to the following JavaScript expression: $(_.user.roles[_bu.id]=='enduser').

   This expression returns a value of true and therefore selects the option by default *only* if the user filling out the form has an enduser role in the current business unit.

■ To select an option by default for request managers only, set the checked attribute of that option to $(_.user.roles[_.bu.id] != 'requestmanager').

■ To select an option by default in a form shared by all business units *only* if the current business unit is PBJ222, set the checked attribute of that option to $(_bu.id=='PBJ222').

■ To select an option by default in a form shared by all business units *only* if the parent of the current business unit is PBJ222, set the checked attribute of that option to $(_bu.id.parent=='PBJ222').

■ To select an option by default if the request was created after a certain date, set the checked attribute of that field to $(new Date(_.request.dateCreated) < 'date').

   For example, to select an option by default if the request was created after January 15, 2010, specify $(new Date(_.request.dateCreated) < new Date().setFullYear(2010,0,15)).

■ To select an option by default if the request status is Approval Done or higher, set the checked attribute of that field to $(_.request.status=>600).

7. Test the JavaScript expression to verify that it accomplishes your goals for enabling or disabling the field on the form.

As a best practice, verify that you test the expression in a form used with a service in a test environment before you use the form and the service in a production environment.

How to Hide, Enable, or Disable an Entire Form Under Specified Conditions

To hide, enable, or disable an entire form under certain conditions, you attach the form to a service option group and enter the JavaScript expression you want in the Disabled and Hidden fields of the Service Option Group Definition dialog. You can   hide, enable, or disable an entire form according to the request status, the role or business unit of the logged in user, or according to other criteria that you specify. For details, see Attach a Form to a Service Option Group (see page 366).

How to Use JavaScript Functions in Fields

CA Service Catalog provides several options for performing automated tasks in form fields, including report data objects, JavaScript expressions, and JavaScript functions. This topic provides an overview of how you may use JavaScript functions for this purpose.

1. If you have not already done so, review the information in How to Perform Automated Tasks for Fields (see page 317) to help decide whether to use a JavaScript *expression* or a JavaScript *function* to perform an automated task for one or fields on a form. If you decide to use a JavaScript function, continue following this topic.

2. Review the pre-defined JavaScript functions (see page 342) and review how to validate the format of commonly requested data (see page 359) to determine how to meet your requirements. Where applicable, review the examples in these topics and optionally use them as models.

3. If none of the pre-defined functions meets your goals, then write your own custom function. As a best practice, store all custom JavaScript functions in the custom_form_lib.js file in the filestore. For information about the filestore, see the *Implementation Guide.*

4. As needed, specify JavaScript functions to link fields in two forms for simultaneous updates (see page 363), so that when a user updates a field in one form, the linked field is automatically updated with the same change.

5. Using all applicable from the previous steps, construct and test the JavaScript function to accomplish your goals for the fields on the form.

6. As a best practice, ensure that you test the function in a form used with a service in a test environment before you use the form and the service in a production environment.

Pre-Defined JavaScript Functions

CA Service Catalog provides several options for performing automated tasks for fields (see page 317) in forms, including report data objects, JavaScript expressions, and JavaScript functions. This topic lists the predefined JavaScript functions and provides an overview of how you can optionally use them to automate tasks in fields.

Use the following guidelines to specify values for the operands used in several of the predefined JavaScript functions. For most functions, the first two operands in the following list are required and the remaining are optional. Verify the syntax for an individual function to see which operands apply to that function.

- For formId, specify the value of the _id attribute of the form on which you want to run the function.

- For _id, specify the value of the _id attribute of the element *within* that form on which you want to run the function.

- For name, value, and index, specify the value of these attributes for the element on which the function is being run.

- Additional operands are mentioned, as applicable, for specific functions. For example, the index attribute is applicable for certain elements only, such as a select group.

_.serviceoption.status( )

Returns the status of the line item to which the current form belongs. Use this function to specify form attributes dynamically depending on the status of the line item. For example, to hide certain form fields when their line status is completed, set their hidden attribute to "_.serviceoption.status() == 200".

Note: You can also use this function to hide or disable a service option element. To do so, use this function in the hidden or disabled text box.

This function is also especially useful for Hold and Resume status. You can give reasons in text messages for both.

Note: This function does not take any parameters.

ca_fdDoFieldLookup(fieldId, reportId),

Runs the report data object, associates it to the lookup field, and copies the data returned to the matching fields on the form. You can use these actions to help populate fields based on user input to a report data object (see page 351).

For fieldId, specify the value of the _id attribute of the lookup field.

For reportId, specify the value of the data object that you created earlier.

ca_reportQuery(reportId, variables, onSuccess, onFailure)

Runs a report data object ([data object](#) (see page 99)) that queries the data source (such as the MDB) for the data you specify and returns the results.

■ For reportId, specify the data object that you want to query.

■ For variables, specify the JavaScript map with the variable names and values. Examples include {"var1":"foo","var2":"bar","var3":_.user.id}. These variables match the ones that you query in the report data object.

■ For onSuccess, specify the JavaScript function to run when the query returns successfully. The onSuccess function must take one operand. This operand is set with an array of maps, each representing a row, returned by the query.

■ For onFailure, specify the JavaScript function to run if the call fails.

You can [prepopulate fields based on a report data object and JavaScript functions](#) (see page 356).

ca_fdValidateCC(credit card number, credit card type)

Validates the format of a credit card number entered by the user in a form you create.

ca_fdShowField(formId, _id)

Applies to a *single* field.

Finds the form (formId) and field (_id) specified. Also makes that field visible if it is not already visible.

ca_fdShowFields(formId, _ids)

Applies to *multiple* fields.

Finds the form (formId) and field (_ids) specified. Also makes the fields visible if they are not already visible.

The second parameter is an array containing the _ids of multiple fields on the form, as shown in the example for ca_fdHideFields(formId, _ids).

ca_fdHideField(formId, _id)

Applies to a *single* field.

Finds the form (formId) and field (_id) specified. Also hides that field if it is not already hidden.

ca_fdHideFields(formId, _ids)

Applies to *multiple* fields.

Finds the form (formId) and fields (_ids) specified. Also hides the fields if they are not already hidden.

The second parameter is an array containing the _ids of multiple fields on the form. For example, you want to hide both the first and last name fields on a form. Also, the _id of the form is name_form, and the _id values of the fields are first_name and last_name. In this case, use the following code:

ca_fdHideFields("name_form", ["first_name","last_name"]

ca_fdDisableField(formId, _id)

Applies to a *single* field.

Finds the form (formId) and field (_id) specified. Also disables the field if it is not already disabled.

ca_fdDisableFields(formId, _ids)

Applies to *multiple* fields.

Finds the form (formId) and fields (_ids) specified. Also disables the fields if they are not already disabled.

The second parameter is an array containing the _ids of multiple fields on the form, as shown in the example for ca_fdHideFields(formId, _ids).

ca_fdEnableField(formId, _id)

Applies to a *single* field.

Finds the form (formId) and field (_id) specified. Also enables the field if it is not already enabled.

ca_fdEnableFields(formId, _ids)

Applies to *multiple* fields.

Finds the form (formId) and fields (_ids) specified. Also enables the fields if they are not already enabled.

The second parameter is an array containing the _ids of multiple fields on the form, as shown in the example for ca_fdHideFields(formId, _ids).

ca_fdSelectOption(formId, _id, name, value) *and* **ca_fdSelectOptionByIndex(formId, _id, index)**

You can use either of these functions to select a value in a select box programmatically. These functions have the same effect, and differ only in how they are called:

■ ca_fdSelectOption selects the option with the specified _id attribute, name, and value in the select field.

■ ca_fdSelectOptionByIndex selects the option with the specified _id attribute and index value in the select field.

Consider the following examples:

Example 1

This example selects the first option in the select box. This example uses the following values:

■ _id attribute of the form: form_id

■ _id attribute of the select box: memory_select

ca_fdSelectOption('form_id', 'memory_select', 1) :

Example 2

This example selects the first option in the select box of the current form. This example uses the following values:

■ _id attribute of the form: ca_fd.formId

■ _id attribute of the select box: memory_select

ca_fdSelectOption(ca_fd.formId, 'memory_select', 1) :

If the specified select box cannot be found, the call is ignored and no errors are reported.

Example 3

You can rewrite the previous examples with explicit values for the options, as follows:

ca_fdSelectOption('form_id', 'memory_select', 'option1', 'option1_value'), ca_fdSelectOption(ca_fd.formId, 'memory_select', 'option1', 'option1_value'):

option1

Specifies the name of an option as it appears on the Forms Designer tree. The value of this option is option1_value.

ca_fdUnselectOption(formId, _id, name, value)

Unselects the option with specified name and value in select field with the corresponding _id attribute.

ca_fdUnselectOptionByIndex(formId, _id, index)

Unselects the option at corresponding index in select field with the corresponding _id attribute.

ca_fdUnselectAllOptions(formId, _id)

Unselects all options in the select field with the corresponding _id attribute.

ca_fdGetSelectedOptions(formId, _id)

Return ans array of integers indicating the indexes of the selected options.

ca_fdGetSelectedOptionValues (formId, _id)

Returns an array of strings indicating the values of the selected options. Select the first option value as follows:

ca_fdGetSelectedOptionValues(formId, _id)[0]

Select the second option value by changing the 0 to 1; select the third option value by changing the 1 to 2, and so forth.

This function is especially useful to

ca_fdSelectRadio(formId, name, _id)

Selects radio with corresponding _id in radio group with corresponding name attribute.

ca_fdIsSelectRadio (formId, name, _id)

Returns whether the indicated radio is selected.

ca_fdSelectCheckBox(formId,_id)

Selects check box with corresponding _id attribute.

ca_fdUnselectCheckBox(formId, _id)

Clears the check box with corresponding _id attribute.

ca_fdIsSelectedCheckBox(formId,_id)

Returns whether the selected checkbox is selected.

ca_fdSetDateFieldValue(formId, _id, date) and related functions

Sets the value of the specified date field with corresponding _id attribute. This function can take Null, String, or Long integer values for the parameter named date.

To clear the value of the field, use empty string and null.

Specify nonempty strings in the format specified for the Date Time field (an element of the form (see page 281)), as follows:

■ If the time portion of the Date Time field is hidden, specify the strings in date format only. Use either the format attribute of the Date Time field or the Date format of the business unit.

■ If the time portion of the Date Time field is visible, specify the strings in the date and time format specified by that field.

In both cases, separate multiple strings with a space.

Related JavaScript functions for setting the date are as follows. The previous text for the ca_fdSetDateFieldValue(formId, _id, date) function also applies to these functions.

■ ca_fdSetDateFieldMinValue(formId, _id, date) – Sets the minimum value of the date.

■ ca_fdSetDateFieldMaxValue(formId, _id, date) – Sets the maximum value of the date.

In addition, for all ca_fdSetDateField* JavaScript functions, you can set the date in the number of milliseconds since the "epoch." The epoch is the standard base time of midnight (00:00:00) GMT, January 1 1970. For details, about setting the date in this manner, see your standard Java programming reference, for example, the Oracle website, oracle.com.

ca_fdGetDateFieldValue(formId, _id, date) and related functions

Gets the value of the specified date field as a string formatted according to the specified formatting for the date field.

The information about the Date Time field for the previous function (ca_fdSetDateFieldValue(formId, _id, date)) also applies to this function.

Related JavaScript functions for getting the date are as follows. The previous text for the ca_fdGetDateFieldValue(formId, _id, date) function also applies to these functions.

- ca_fdGetDateFieldValueInMillis(formId, _id, date) – Gets the value of the date in milliseconds.

- ca_fdGetDateFieldMinValue(formId, _id, date) – Gets the minimum value of the date.

- ca_fdGetDateFieldMinValueInMillis(formId, _id, date) – Gets the minimum value of the date in milliseconds.

- ca_fdGetDateFieldMaxValue(formId, _id, date) – Gets the maximum value of the date.

- ca_fdGetDateFieldMaxValueInMillis(formId, _id, date) – Gets the maximum value of the date in milliseconds.

In addition, for all ca_fdGetDateField* JavaScript functions, you can get the date in the number of milliseconds since the "epoch." The epoch is the standard base time of midnight (00:00:00) GMT, January 1 1970. For details, about getting the date in this manner, see your standard Java programming reference, for example, the Oracle website, oracle.com.

ca_fdSetTextFieldValue(formId, _id, text)

Sets text of text field with corresponding _id attribute, which can be text field and text area.

ca_fdGetTextFieldValue(formId, _id)

Gets text of text field with corresponding _id attribute, which can be text field and text area.

ca_fdFetchSelectData(formId,_id)

Causes the select box with the _id specified to fetch the report data object again. This function is especially useful when the report data object takes user input into account. As a best practice, invoke this function when that input changes, to help ensure that the select box displays the correct data.

This function is also especially useful to use user input to prepopulate a select box (see page 322).

The following functions are for system forms (see page 377) *only*:

- custom_onPriorityChange(newPriority)

- custom_onDateRequiredChange(newDateRequired)

- custom_onRequestedForChange(type, id, name, shippingAddress)

Predefined JavaScript Functions for Tables Only

The following predefined JavaScript functions apply to tables only:

ca_fdAddTableRow(formId, tableId, {}):

Adds a row to the table.

Object attributes correspond to the _id of the table columns. The values of the attributes correspond to the value of the cell in the row.

The third argument contains the row data as a JavaScript object, for example, {'userid':'spadmin','age':'31'}.

The following call adds a row with cells named spadmin and 31:

```
ca_fdAddTableRow('myForm','myTable',{'userid':'spadmin','age':'31'})
```

The newly added row is automatically selected. The table contains two columns whose _id values are userid and age.

ca_fdRemoveTableRow(formId, tableId, {}):

Removes the specified row from the table.

Note: This function removes *only* rows that users added while editing the form during a request. This function does *not* remove rows that an administrator populated while creating a dynamic table (see page 290) or static table (see page 288).

The values of the object attributes correspond to the values of the _id attributes of the cell in the row.

The third argument contains the row data as a JavaScript object, for example, {'userid':'spadmin','age':'31'}. This argument is comma-separated list of key, value pairs, where:

- The key is the _id attribute of the column.

- The value is the value for the column.

Note: Keys and values do not need to be string constants, but can be JavaScript expressions that evaluate to strings.

For example, the following call removes a row with cells named spadmin and 31:

```
ca_fdRemoveTableRow('myForm','myTable',{'userid':'spadmin','age':'31'})
```

ca_fdRemoveAllTableRows(formId, tableId):

Removes all rows from the table, except the rows that were dynamically or statically populated, as explained in the note for the previous item, ca_fdRemoveTableRow(formId, tableId, {}).

The following call removes all user-added rows from the specified table:

```
ca_fdRemoveAllTableRows('myForm','myTable')
```

ca_fdFetchTableData(formId, tableId):

Reruns the report data object associated with the table and refreshes the data in the table.

This function is useful to refresh data based on a change in user input, for example, if *both* of the following are true:

■   You are using a report data object to populate the table based on user input.

■   The user changes a selection in a <u>select box</u> (see page 287), meaning that the data must be refreshed.

Note: This function is valid only when you use a report object attribute to populate the table.

ca_fdGetTableRowCount(formId, tableId):

Returns the number of rows in the table.

ca_fdGetTableSelectedRowCount(formId, tableId):

Returns the number of selected rows in the table.

ca_fdGetSelectedTableRows(formId, tableId):

Returns all selected table rows as JavaScript objects.

Examples

The following are sample JavaScript functions for tables, for illustration. In these examples, the following apply:

■   The value of the _id attribute of the table is *t*.

■   The values of the _id attributes of the columns are *label* and *sf*.

```
ca_fdSetTableRow(this.formId, 't', 0, {'label' : 'new row label', 'sf' : '3'})
```

```
ca_fdFetchTableData(this.formId, 't')
```

```
ca_fdGetTableRowCount(this.formId, 't')
```

```
alert(JSON.stringify(ca_fdGetSelectedTableRows(this.formId, 't')))
```

```
ca_fdUnselectTableRowsByIndex(this.formId, 't', [0 ])
```

```
ca_fdUnselectTableRows(this.formId, 't', [{ label text", sf)
```

```
ca_fdSelectTableRowsByIndex(this.formId, 't', [0,1])
```

How to Populate Fields Based on User Input to a Report Data Object

CA Service Catalog provides several options for using JavaScript functions to populate fields on a form based on data entered by the user. This topic explains one of these options. Specifically, this topic explains how to populate fields based on user input, by using a lookup field, a report data object (data object), and the pre-defined JavaScript function named ca_fdDoFieldLookup(fieldId, reportId). You use them together to run a data object and return the results in the matching field or fields on the form.

When the user clicks the magnifying glass icon for the lookup field on the form, CA Service Catalog prompts the user to enter the variable that you specified in your data object. An example is the user ID.

When the user answers the prompt or prompts in the lookup field, CA Service Catalog runs the data object and searches the data source for the data requested in fields of the data object; for example, the first name and last name of the user ID entered in the lookup field. In this example, you create matching First Name and Last Name fields on the form, and CA Service Catalog returns the results to the user. The user selects which result (if any) to use to populate the form fields.

To populate fields based on user input, by using a lookup field, a data object, and the pre-defined JavaScript function named ca_fdDoFieldLookup(fieldId, reportId), follow these steps:

1. In the Report Builder, create a data object (see page 99) that queries the data source (such as the MDB) for the data you want (the data copied to the matching fields on the form). The query is based on the user input to the prompt or prompts that you specify, such as user ID.

   Record the name of the data object for later reference.

   The following sample query for the MDB matches the example started earlier in this topic:

   SELECT userid,first_name,last_name FROM ca_contact WHERE userid = '%userid%'

   The ca_fdDoFieldLookup function processes this query as follows:

   ■ "WHERE userid = '%userid%'" specifies that the user is prompted to enter this value, and this value is used in the SELECT clause.

   ■ "SELECT userid,first_name,last_name" specifies which values to return.

   ■ "FROM ca_contact" specifies that the data source is the ca_contact table in the MDB.

2. On the form, add the lookup field; it is one of the elements of a form (see page 281).

    a. For the _id attribute, specify the first field returned by the query, and save the form.

       Thus, in the continuing example, you specify *userid* as the value of the _id attribute.

    b. For the value of the JavaScript attribute (see page 315) named onLookup, specify the pre-defined JavaScript function (see page 342) named ca_fdDoFieldLookup(fieldId, reportId) and save the form.

       This function runs the data object, associates it to the lookup field, and copies the data returned by the data object to the matching fields on the form.

       For fieldId, specify the value of the _id attribute of the lookup field.

       For reportId, specify the value of the _id attribute of the data object that you created earlier.

    c. For the tool tip attribute, optionally specify instructional text, such as "Click the magnifying glass and enter the requested data." Save the form.

       In the continuing example, for the tool tip attribute, you may optionally specify text such as "Click the magnifying glass and enter user ID."

    d. Optionally rename the default display text from the lookup field from "Lookup Field" to a more meaningful name, by selecting the element in the Component Tree and clicking the Rename icon at the top of the tree.

       In the continuing example, you may rename the display text to "User ID" or something similar.

    e. Leave value attribute empty. This value is populated with the first result returned by the query. The value appears on the form to the user as the value of the lookup field.

3. Also on the form, typically under the lookup field, add the fields that hold the results returned by the query. For each field, specify the value of the HTML Attribute (see page 303) named _id to match exactly the ID attribute of the corresponding object in the database.

   Note: To optionally disable a field so that users cannot change the result returned by the query, set the disabled attribute of the field to true. You can also set the disabled attribute to true only under certain conditions that you specify.

   In the continuing example, create matching First Name and Last Name fields on the form, and specify   first_name and last_name, respectively, in the _id attributes of these fields.

   Thus, all matching values of userid are returned in the lookup field; each result appears as a row in the search results. Users can review the search results and select a row. When the user selects a row, the lookup field closes, and the form is populated with the results from that row. The user sees something like the following on the form:

   User ID: johsmi515

   First Name: John

   Last Name: Smith

4. Test the form to verify that it works as you intended.

   Important! The lookup field displays *only* the first 25 matches in the search results. If your query returns more than 25 matches, refine your query to return fewer matches.

   As a best practice, verify that you test the expression in a form used with a service in a test environment before you use the form and the service in a production environment.

Use User Input for Personal Data to Pre-Populate a Select Box

You can use the first few characters provided by the user in a select box in a form to trigger an auto-complete function. A common application of this technique occurs on a form with a select box (an element of a form (see page 281)) for the last name. For example, if the user types the letters *smit*, the field displays all valid options, such as Smittapopolous, Smitderski, Smitapunangala, and so forth. The user can select a valid value but cannot enter a custom value. This technique helps users find and specify valid data efficiently when the database is large.

Follow these steps:

1. In the Report Builder, create a data object (see page 99) that queries the data source (such as the MDB) for the data you want. The query must return the list of all possible values and populate the select box with this list. This step is part of the process of pre-populating a combo box based on a report data object (see page 320). For this example, do the following:

   ■ Create a report data object to retrieve the list of user IDs from the database. When the form appears, the report data object runs and populates the combo box with the resulting data.

   ■ Record the ID of the data object for later reference.

   ■ Verify that the variable you want to use exists and is of type *string*. If necessary, create or update the variable. For illustration, we use the last_name_prefix variable in the following example.

   For example, consider the following sample query for a data object for the MDB. This example returns the list of available last names that match the user input in the field.

   Select userid as id, last_name from ca_contact where last_name like '%last_name_prefix%%%'

   Select userid as id, last_name

   Specifies which values to return. It returns the ID and the last name of all users that match the first few characters entered by the user.

   From ca_contact

   Specifies the name of the database table.

   where last_name like '%last_name_prefix%%%'

   Specifies the filter clause.

   The last_name_prefix variable filters on the last_name column (the display column of the query).

   The 'like '%last_name_prefix%%%' syntax filters the results so that the last_name starts with last_name_prefix.

2. Add the select box to the form.

   Note: When you use a data object to populate a select box, do not add any options to it, because they are ignored (not used). When the user opens the form, any options for the select box are ignored, only the data object is used.

   a. For the _id attribute, specify a meaningful value, and save the form.

   b. (Optional) Rename the default display text from the select group from "Select" to a more meaningful name, as follows: Select the element in the Component Tree and click the Rename icon at the top of the tree.

3. Specify values for the following attributes of the select box, and save the form.

   ■ Multi-Select: Specify False so that users can select only one option in the combo box.

   ■ (Optional) Title (tooltip text): Specify instructional text, such as "Click the arrow and scroll to select a last name.

     Note: You can optionally [localize](see page 375) the tooltip text or the name of the select box, if applicable.

4. Specify values for the following attributes of the select box, and save the form.

   ■ Report/Plug-in Id: Specify the name of the report data object that you created in Step 1.

   ■ Report/Plug-in Variable: Specify a JSON expression for the object that maps a key value pair, as follows:

     – The key is the variable for the report data object.

     – The value is a JavaScript expression that matches the data you want.

     For this example, specify the following:

     `$({'last_name_prefix':_val})`

     – last_name_prefix is a string constant, as noted in Step 1.

     – _val is a predefined variable that CA Service Catalog provides.

   ■ Minimum Search Characters: Specify a custom value, for example, 4 or 5. This attribute specifies the number of characters that users must type to trigger the auto-complete feature. The defaults is 4.

5. Test the form to verify that it works as you intended.

As a best practice, test this form with a service in a test environment before you use them in a production environment.

How to Pre-Populate Fields Based on a Report Data Object and JavaScript Functions

To pre-populate fields with data that is not accessible through JavaScript expressions (see page 325), you can use a report data object (data object), custom JavaScript functions, and pre-defined JavaScript functions (see page 342), especially ca_reportQuery(reportId, variables, onSuccess, onFailure). A representative case involves querying your organization's Human Resources database (not the CA MDB) for sensitive data, such as the banking account number and related information for a user.

For example, you may create a form for changing the bank account to which a user's pay check is automatically deposited. In this form, you may pre-fill fields with name of the current bank name and account number. Use later fields on the form for the user to enter the name of the new bank name and account number.

First, ca_reportQuery(reportId, variables, onSuccess, onFailure) retrieves from the data object the variables that match the fields queried by the data object. Next, the OnSuccess function is called, at which point you can use other pre-defined JavaScript functions (see page 342) to pre-populate multiple fields in the form with the results of the query. For example, you may write a custom function that calls ca_fdSetTextFieldValue(formId, _id, text) multiple times, once for each field you want to pre-populate. This scenario is typically most suitable for pre-populating text fields, but can also be used to populate other types of fields.

To pre-populate data in form fields using a data object and both custom and pre-defined JavaScript functions, follow this process:

1. Design and create the form (see page 296), if you have not already done so.

2. In the Report Builder, create a data object (see page 99) that queries the data source for the data you want (the data to be copied to the matching fields on the form).

   Record the ID of the data object for later reference.

   The following sample query for the MDB matches the example started earlier in this topic:

   SELECT bank_name,account_number FROM my_hr_db WHERE userid ='%userid% '

   CA Service Catalog processes this query as follows:

   a. "WHERE userid = '%userid% ' specifies that a user id must be provided as input to this data object.

   b. "SELECT bank_name,account_number" specifies which values to return.

3. "FROM my_hr_db" specifies that the database is my_hr_db, a fictitious database for illustration.

4. On the form, add text fields (or other suitable form elements instead) to receive the retrieved data, as follows:

   ■ Add one text field for each variable queried by the data object.

     For the continuing example, add two text fields, one for bank name and one for account number.

     For the continuing example, add two text fields, specify bank_name and account_number, respectively.

   ■ Rename each text field to have a meaningful name.

     For the continuing example, rename the fields to Bank Name and Account Number, respectively.

5. Select the form name to display the form attributes.

6. In the form attributes, in the JavaScript attribute named onload, enter the name of your function that calls ca_reportQuery. For illustration, these steps use getAcctData() as the name for this function. Substitute the name of your function if it is different.

7. Verify that getAcctData() is declared in a JavaScript file accessible by CA Service Catalog.

   For best results, use this location in the filestore: FileStore/scripts/custom_form_lib.js.

   Important! The folder name *FileStore* is case sensitive. Therefore, use the correct case in path names and all other programmatic references.

   Note: For information about setting up the filestore, see the *Implementation Guide*.

8. In the body of getAcctData, specify the call to ca_reportQuery function, to run the data object and return the results.

9. Use the results as input values to JavaScript functions such as ca_fdSetTextFieldValue to update the fields in the form.

   In this example, use the results to make two calls: one to update the bank name field and another to update the account number field.

10. Customize ca_reportQuery(reportId, variables, onSuccess, onFailure) to meet your requirements. This step uses the continuing example for illustration.

    ■ For reportId, specify the data object that you created earlier.

    ■ For variables, specify the JavaScript map with the variable names and values.

      For the continuing example, specify: {"userid":_.user.id}.

      These variables *must* match the ones that you query in the data object, but they do not need to be ordered in the same sequence.

■   For onSuccess, specify the custom JavaScript function to run when the query returns successfully. The onSuccess function must take one operand. This operand is set with an array of maps, each representing a row, returned by the query. Within your custom onSuccess function, use ca_fdSetTextFieldValue(formId, _id, text) to return the results of each field, one at a time.

For the continuing example, specify the following:

```
function updateFields(result) {
    if (result.length == 1) {
    ca_fdSetTextFieldValue(formId, 'bank_name', result[0].['bank_name']);
    ca_fdSetTextFieldValue(formId,'account_number',result[0].['account_number']);
    } else {
        alert('could not find your bank account information')
    }
}
```

■   For onFailure, specify the JavaScript function to run if the query fails.

For the continuing example, specify the following:

```
function onGetAcctDataFail() { alert("can't find your account"); }
```

11. Finish creating the form to meet your requirements.

12. Test the form to verify that it works as you intended.

As a best practice, verify that you test the expression in a form used with a service in a test environment before you use the form and the service in a production environment.

How to Validate User Input

You can configure Form Designer forms to verify that users entered the following types of data in the correct format: credit card numbers, social security numbers, email addresses, phone numbers, postal (zip) codes, and decimal numbers with two fractional digits.

This process does *not* guarantee that the data entered by the user is authentic but does assist in the validation process by verifying that the user entered the data in the correct format.

You can thus use the Form Designer to verify that the data you want and is recorded in the format that you want. For example, you can verify that phone number fields accept only numeric input, that date fields accept only your company's standard date format, and that credit card fields accept only numbers that are entered in the correct format for the credit card type specified.

Additionally, you can specify maximum length, minimum length, and other attributes used to validate user input.

Choose the task you want to perform:

- Use regular expressions to validate numeric and address data (see page 361). In this case, you use regular expressions in the HTML attribute (see page 303) named pattern to validate the format of credit card numbers, social security numbers, email addresses, phone numbers, IP addresses, postal codes, dates, and decimal numbers with two fractional digits.

- Use JavaScript functions to validate the format of credit card numbers. For example, you use the pre-defined JavaScript function (see page 342) named ca_fdValidateCC(credit card number, credit card type) in a JavaScript attribute (see page 315) such as onvalidate to verify the credit card formats. Alternatively, you can create your own custom JavaScript function.

How to Use a JavaScript Function to Validate the Format of Credit Card Numbers

This topic explains how to configure Form Designer forms to use a JavaScript function to verify that users entered a credit card number in the correct format for the credit card type that they specified. This process does *not* authenticate the data entered by the user but does assist in the validation process by verifying the format.

Note: You can also use regular expressions (see page 361) to validate the format of credit card numbers and other kinds of commonly requested data, such as social security numbers, email addresses, phone numbers, and so forth.

This topic explains how to present two credit card types: Master Card and Visa. For each type, you create a radio group option and a text field for the account number. Each account number field is disabled by design but is enabled if the user selects the matching credit card type.

You use the pre-defined JavaScript function (see page 342) named ca_fdValidateCC(*number*, '*type*') to validate the format of the credit card number entered by the user, based on the credit card type that the user selected. This JavaScript function verifies the format according to the specifications set by the company issuing the credit card. Each company sets its own standard format.

To verify the credit card numbers entered by users, follow these steps. These steps use a continuing example for illustration.

1. Design and create the form (see page 296), if you have not already done so.

2. Verify that the value of the form's HTML Attribute (see page 303) named _id specifies a meaningful name, such as ccValdtnForm1.

3. Add a radio group; it is one of the elements of a form (see page 281).

    a. For the value of the _id attribute of the radio group, specify a meaningful name, such as rgCCVal, and save the form.

    b. Rename "Radio Group" to a meaningful name or phrase, such as Select a Credit Card or Credit Card Type.

4. Add the first radio group option; these options are elements of a form (see page 281) that apply only to radio groups.

    a. For the _id attribute, specify a meaningful name, such as mcard (signifying Master Card radio group option), and save the form.

    b. Rename "Radio" to a meaningful name, such as Master Card.

5. Add the second radio group option.

    a. For the _id attribute, specify a meaningful name, such as visa (signifying Visa radio group option), and save the form.

    b. Rename "Radio" to a meaningful name, such as Visa.

6. Add a text field onto the form

    a. For the _id attribute, specify a meaningful name, such as ccf.

    b. For the value of the JavaScript attribute (see page 315) named onvalidate, specify cc_val(_val).

7. In your custom JavaScript library, add the following function definition.

    Note: As a best practice, use custom_form_lib.js in the filestore as your custom JavaScript library. For details about setting up the filestore, see the *Implementation Guide*.

    ```
    function cc_val(value) {
                    var ccname = '';
                    if (ca_fdIsSelectRadio('ccValdtnForm1', 'rgCCVal', 'mcard'))
                                    ccname = 'master';
                    else
                                    ccname = 'visa';
                    ca_fdValidateCC(value, ccname);
    }
    ```

8. Finish creating the form to meet your requirements.

9. Test the form to verify that it works as you intended.

    As a best practice, verify that you test the form in a service in a test environment before you use the form and the service in a production environment.

How to Use Regular Expressions to Validate Numeric and Address Data

This topic explains how to use regular expressions to configure Form Designer forms to verify that users entered the following types of data in the correct format: credit card numbers, social security numbers, email addresses, IP addresses, phone numbers, postal (zip) codes, and decimal numbers with two fractional digits. This process does *not* verify that the data entered by the user is authentic but does assist in the validation process by verifying that the format is correct.

Note: You can also use JavaScript functions to validate the format of credit card numbers.

To verify the types of user input mentioned earlier, follow these steps. These steps use a continuing example for illustration.

1. Design and create the form (see page 296), if you have not already done so.

2. For each type of user input, add a text field.

3. Rename each text field from Text Field to a meaningful name, such as Social Security Number, Credit Card Number, and so forth.

4. For the value of the HTML Attribute (see page 303) named pattern, specify the regular expression for the type of data being validated, and save the form. Valid values follow:

   - Social security number: ^([0-6]\d{2}|7[0-6]\d|77[0-2])([\\d{2})\2(\d{4})$

   - Decimal number with two fractional digits: ^[1-9]\d*(\.\d{2})$

   - E-mail address:
     ^([0-9a-zA-Z]+([_.-]?[0-9a-zA-Z]+)*@[0-9a-zA-Z]+[0-9,a-z,A-Z,.,-]*(.){1}[a-zA-Z]{2,4})+$

   - IP Address:
     ^((25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.){3}(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})$

   - Telephone number: ^(([0-9]{1})*[- .(]*([0-9a-zA-Z]{3})*[- .)]*[0-9a-zA-Z]{3}[-.]*[0-9a-zA-Z]{4})+$

   - URL:
     ^(http[s]?://|ftp://)?(www\.)?[a-zA-Z0-9-\.]+\.(com|org|net|mil|edu|ca|co.uk|com.au|gov)$

   - Postal Code (CAD): ^([A-Z][0-9]){3}$

   - For additional values, or to create your own expression, see your reference for regular expressions, such as www.regular-expressions.info or www.regexlib.com.

5. For the value of the HTML Attribute (see page 303) named patternMessage, specify an error message to appear when the user input violates the pattern attribute.

   Note: You can optionally localize (see page 375) this error message.

6. Finish creating the form to meet your requirements.

7. Test the form to verify that it works as you intended.

As a best practice, test the regular expression in a form used with a service in a test environment before you use the form and the service in a production environment.

How to Link Fields in Two Forms for Simultaneous Updates

You can optionally specify JavaScript functions to link two fields in two Form Designer forms. When a user updates a field in one form, the linked field is updated in the second form simultaneously. This technique essentially creates an automatic cross-reference between the fields in both forms. To use this technique, follow this process:

1. Verify that the forms are in the same service option.

   Important! The forms *must* be in same service option.

2. Decide which fields in which forms you want to link. For example, you may want to ensure that a name, date, item, or phone number field has the same value in both forms in each request.

3. Specify the same exact function in both forms *except* that you switch the form IDs in each JavaScript function, as follows:

   ■ In the JavaScript function in Form 1, you specify the ID for Form 2.

   ■ In the JavaScript function in Form 2, you specify the ID for Form 1.

   ■ For ease of maintenance, use the same field name and same JavaScript attribute in each form.

     For example, you may decide to name the field Address in both forms and specify the JavaScript function in the onclick attribute in each form.

Important! Do *not* re-use either form in a different service. If one of the forms is re-used in a different service, the fields become un-linked. Consequently, both fields function correctly individually but are no longer cross-referenced. Therefore, they are no longer configured for simultaneous updates.

Example

This example illustrates how to use JavaScript functions to link two fields in two Form Designer forms for simultaneous updates.

In Form AA, whose form ID is form_aa, specify the following for the linked field:

■ Name: Item Requested

■ JavaScript function in the onclick attribute:
  ca_fdSetTextFieldValue('form_bb','item_requested','New Laptop')

In Form BB, whose form ID is form_bb, specify the following for the linked field:

■ Name: Item Requested

■ JavaScript function in the onclick attribute:
  ca_fdSetTextFieldValue('form_aa','item_requested','New Laptop')

Guidelines for Custom JavaScript Functions

Use the following guidelines for creating custom JavaScript functions:

- Follow the instructions for using pre-defined JavaScript functions (see page 342), as applicable.

- Follow the instructions for using JavaScript attributes (see page 315).

- As a best practice, use the Script dialog (see page 365) on each form the store the custom JavaScript functions that apply to the form. Storing each custom function in the Script dialog enables you to more efficiently move the form, custom functions, and related services or service option groups together. This consideration can be important if you move these objects because of an upgrade, migration, or other event.

  Note: To move these objects, use the ixutil command. For details, see the *Reference Guide.*

  Alternatively, store custom JavaScript functions in the custom_form_lib.js file in the filestore.

  Note: For information about the filestore, see the *Implementation Guide.*

- Design your forms with usability as a high priority.

- Do not write excessively long custom JavaScript functions.

- Take into account the browser locale, to support other language users and to localize forms (see page 375) as needed.

Use the Script Dialog to Maintain Custom JavaScript Functions

The Script dialog for each form lets you create and maintain JavaScript functions that apply to a form. Using the Script dialog is the best-practice option (see page 364) for maintaining custom JavaScript functions for a form. The functions that you save in the dialog apply *only* to the form that you are editing. To use the Script dialog for a form, you require access rights to edit the form.

Follow these steps:

1. Click the name of the form in the Form Designer tree.

   The form attributes appear, and the Script button activates. The Script button appears on the toolbar with the Localize, Save, Restore, and other buttons.

2. Click the Script button.

   The Script dialog appears and displays the custom JavaScript functions already specified for the form, if any. You can enter new JavaScript functions before or after any existing ones.

   Note: To view a template for a function, click Show Help on the Script dialog. The template shows the product-specific implementation of a standard JavaScript function.

3. Write and save the custom function. Use the format shown in the sample functions that appear when you click Show Help on the Script dialog. This format is known as *JavaScript object literal notation*.

4. Save the form.

5. Specify the name of the function in the JavaScript attribute (see page 315) from which you want to call it. Use the following format:

   ca_fd.js.*function-name*

   An example follows:

   ca_fd.js.calculateCostQty()

   Typically, you specify a custom JavaScript function in one of the following:

   ■ The onSubmit or on Load attribute of the form attributes (see page 297)

   ■ The onChange or onValidate attributes of applicable fields

You have used the Script dialog to maintain JavaScript functions that apply to a form.

Attach a Form to a Service Option Group

You must attach a form to one or more applicable service option groups, so that end users who select one of these service options in a request are prompted to complete the form. The form is a mechanism for the user to supply information needed to complete the request to CA Service Catalog.

To attach a form to a service option group

1. Click Service Builder, Service Option Groups from the main menu.

   The Service Option Groups appear.

2. Click the service option group name to which you want to add the form.

   The Service Option Group Definition window appears.

3. Double-click the new row or column heading to add a new row or column of cells, as follows:

   ■ Double-click a new, empty cell (except the first cell) to define a new service option element.

   ■ Double-click an existing service option element to edit the definition of an existing service option element.

   The Service Option Element Definition window appears.

4. Click the Definition tab.

5. In the Type drop-down list, select Form; do *not* select CA-Workflow form.

   The form-related options appear on the tab.

6. In the Display Text field, enter meaningful text that describes the purpose of this form.

   This text appears to the user as a description for the form when administrators view this service option group in the Service Builder.

7. In the Form Name field, click the magnifying glass icon to select a form.

   The Form Tree pop-up window appears.

8. Navigate through this form tree and select the form that you want to attach.

   The preview of the form appears in the right pane.

9. Click the Select Form button.

   Note: You can add a form to a service option only once. In other words, you cannot add the same form to the same service option multiple times.

10. If you do *not* want to hide or disable the entire form at any point for any reason, leave the Hidden and Disabled fields blank, and skip the remainder of this step.

    If you *do* want to hide or disable the entire form according to the request status, the user's role or business unit, or other criteria, enter the corresponding JavaScript expression in the Hidden or Disabled field. Use the following format: $(_.*object*.*property*). The expression must return a value of true or false.

    You can specify a JavaScript expression in the Hidden field, the Disabled field, or both. For instructions to specify a JavaScript expression correctly, see How to Use JavaScript Expressions in Fields (see page 325). That section explains how to specify expressions in fields *within* a form; those instructions also apply to any expressions that you specify for the *entire* form in the Disabled and Hidden fields in this step.

    Examples follow:

    ■   To hide or disable the form when the request status is Pending Approval, enter the following JavaScript expression in the Hidden or Disabled field: $(_.request.status == 400).

    ■   To hide or disable the form for end user roles *only*, enter $(_.user.role == 'enduser').

    ■   To hide or disable the form from all business units except ca.com, enter $(_.bu.id != 'ca.com').

    ■   To disable the form when the request status is Fulfilled, enter $(_.request.status == 2000).

    Note: When an entire form is disabled, it is disabled but visible during all stages of the request life cycle *except* check-out. During check-out, a disabled form is both disabled and hidden.

11. Click the Options tab.

    The remaining service option element fields appear.

12. Complete the remaining fields on the Service Option Element Options Window--Options Tab (see page 196) and click Update.

How to Create and Maintain Folders

You create and maintain folders in the Form Designer similar to the way that you create and maintain folders in the Service Builder. In the Form Designer, you may want to create and maintain folders to help you organize your forms according to groups, categories, or business units. For example, you may create and name folders that correspond to each business unit, department, owner, or service. Thus, you may create folders and sub-folders for Business Unit 1, Business Unit 2, Accounting, Finance, Customer Service, Human Resources, and so forth.

To create and maintain folders, follow this process:

1. Verify that you have administrator rights to each folder that you want to access, in each affected business unit. For example, to create or move a folder in Business Unit 12, you must be defined as an administrator in that business unit. Similarly, to copy or move a folder from Business Unit 1 to Business Unit 5, you must be defined as an administrator in both business units. If you are an administrator in a parent business unit, then you are automatically an administrator in all of its children business units.

   Note: You cannot create, copy, move, or delete any folders or elements within the System folder.

2. Use caution when modifying an existing folder. If the folder contains any forms or elements that are included in an *active* request, you receive a warning message.

   Do *not* make any changes (except trivial changes) to the folder or its contents if you receive this warning message. When you no longer receive this message, proceed with your change.

3. Use *extreme* caution when deleting an existing folder or moving an existing folder from one business unit to another. If the folder contains any forms or elements that are included in an *active* request, you receive a warning message. Do *not* complete the move or delete operation if you receive this warning message. When you no longer receive this message, proceed with your change.

   Important! When you delete a folder, all sub-folders, forms, and elements under it are also deleted.

4. Click Service Builder, Form Designer.

   The Form Designer window appears; the Components and Forms folders are collapsed in the Component Tree (the left pane) and the Form Design Area (the middle pane) blank.

5. In the Component Tree, click the icon next to the Forms folder to expand it.

   The default folder and any custom folders that you have already created appear under the Forms folder.

6. To create a folder, do the following:

   a. To create the folder on same branch as the folders that currently appear, click Add and select Folder from the drop-down menu.

   b. To create the folder on a lower branch than the folders that currently appear, expand the tree as needed to display the branch at which you want to create the new folder, click Add, and select Folder from the drop-down menu.

      The Add Folder dialog appears.

   c. Enter the name of the new folder and click OK. The name must be unique within its business unit.

      The new folder is created. You are now ready to begin adding forms or sub-folders to the new folder.

7. To rename a folder, do the following:

   a. To rename a folder on same branch as the folders that currently appear, select the folder and click Rename.

   b. To rename a folder on a lower branch than the folders that currently appear, expand the tree as needed to display the branch that contains the folder that you want to rename, and click Rename.

      The Name Folder dialog appears.

   c. Enter the new name of the new folder and click OK. The name must be unique within its business unit.

   The folder is renamed.

8. To move a folder, do the following:

   a. Expand the tree to display the folder being moved.

   b. Expand the tree to display the destination folder (the folder to which you are moving the original folder).

   c. Select the folder being moved and click Cut.

   d. Select the destination folder and click Paste.

   The folder is moved.

   Note: If the folder was moved from one business unit to another, the moved folder, including all sub-folders, forms, and elements under it, now belong to the business unit of the destination folder. The names of each folder, form, and element must be unique within their new business unit.

9. To copy a folder, do the following:

   a. Expand the tree to display the folder to be copied.

   b. Expand the tree to display the destination folder (the folder under which you are copying the original folder).

   c. Select the folder being copied and click Copy.

   d. Select the destination folder and click Paste.

   The folder is copied, and its new name is "copy of *original name.*" However, the sub-folders, forms, and elements under it are *not* renamed.

   Note: If the folder was copied from one business unit to another, the pasted folder, including all sub-folders, forms, and elements under it, now belong to the business unit of the destination folder.

   e. Optionally, rename the newly copied folder and the sub-folders, forms, and elements under it.

10. To delete a folder, do the following:

   a. Expand the tree to display the folder to be deleted.

   b. Select that folder.

   c. Click Delete.

   The folder is deleted. All sub-folders, forms, and elements under it are also deleted.

How to Maintain Forms

You create and maintain forms in the Form Designer similar to the way that you create and maintain folders in the Form Designer. In the Form Designer, you may want to create and maintain folders to help you organize your forms according to groups, categories, or business units. After doing so, you may later edit the forms. You may also move or copy forms from one folder to another, and these operations may affect two or more business units, departments, owners, or services. For example, you may move one or more forms from a folder in Business Unit 1 to another folder in Business Unit 5. Similarly, you may copy a form from Accounting to Finance, from Customer Service to Human Resources, and so forth.

To edit, move, copy, and delete forms, follow this process:

1. Verify that you have administrator rights to each folder that you want to access, in each affected business unit. For example, to create or edit a form in Business Unit 12, you must be defined as an administrator in that business unit. Similarly, to copy or move a form from Business Unit 1 to Business Unit 5, you must be defined as an administrator in both business units.   If you are an administrator in a parent business unit, then you are automatically an administrator in all of its children business units.

   Note: You cannot create, copy, move, or delete any forms or elements within the System folder.

2. Use caution when modifying an existing form. If the form is included in an *active* request, you receive a warning message.

   Do *not* make any changes (except trivial changes) to the form if you receive this warning message. When you no longer receive this message, proceed with your change.

   As a best practice, if you need to make important changes to the form, create a new version with the changes and replace the old form with the new form.

3. Use extreme caution when deleting an existing form or moving an existing form from one business unit to another. If the form is included in an *active* request, you receive a warning message. Do *not* complete the move or delete operation if you receive this warning message. When you no longer receive this message, proceed with your change.

   Important! When you delete a form, all elements under it are also deleted. If you want to save the elements, store them in another form before deleting this form.

4. Click Service Builder, Form Designer.

   The Form Designer window appears; the Components and Forms folders are collapsed in the Component Tree (the left pane) and the Form Design Area (the middle pane) blank.

5. In the Component Tree, click the icon next to the Forms folder to expand it.

   The default folder and any custom folders that you have already created appear under the Forms folder.

6. [Create the form](#) (see page 296), if applicable.

7. To edit a form, do the following:

   a. Open the folder that contains the form you want to edit.

   b. Open the form to display its existing elements.

   c. [Add, modify, and configure elements on the form](#) (see page 298).

   The form is updated.

8. To rename a form, do the following:

   a. To rename a form on same branch as the folders that currently appear, select the form and click Rename.

   b. To rename a folder on a lower branch than the folders that currently appear, expand the tree as needed to display the branch that contains the folder that you want to rename, select the form, and click Rename.

      The Name Form dialog appears.

   c. Enter the new name of the new form and click OK. The name must be unique within its business unit.

   The form is renamed.

9. To move a form, do the following:

   a. Expand the tree to display the form.

   b. Expand the tree to display the destination folder (the folder to which you are moving the form).

   c. Select the form being moved and click Cut.

   d. Select the destination folder and click Paste.

   The form is moved.

   Note: If the form was moved from one business unit to another, the moved form, including all elements under it, now belong to the business unit of the destination folder. The name of the moved form must be unique within their new business unit.

10. To copy a form, do the following:

    a. Expand the tree to display the form.

    b. Expand the tree to display the destination folder (the folder to which you are copying the original form).

    c. Select the form being copied and click Copy.

    d. Select the destination folder and click Paste.

       The form is copied, and its new name is "copy of *original name.*" The copied elements under the copied form are *not* renamed.

> **Note:** If the form was copied from one business unit to another, the pasted form, including all elements under it, now belong to the business unit of the destination folder.

   e.   Optionally, rename the newly copied forms.

11. To delete a form, do the following:

   a.   Expand the tree to display the form.

   b.   Select the form.

   c.   Click Delete.

The form is deleted. All elements under it are also deleted.

Form Designer Access Level Rules

Users with the following roles (authorized users) can access forms: Super Business Unit Administrator, Catalog Administrator, Service Manager, and Service Delivery Administrator. Here, *access forms* means that they can view, create, edit, copy, and delete forms, unless noted otherwise in the following bullets.

Administrators can configure access rights to Form Designer forms by using the configuration settings named Use Service Provider Catalog Only and Pass Through Catalog, as follows:

■ To access and optionally change the Use Service Provider Catalog Only setting, log in to the root (highest level) business unit as a Service Delivery administrator and select Service Builder, Configuration, System Configuration.

■ To access and optionally change the Pass Through Catalog setting, log in to the business unit of interest as an authorized user and select Service Builder, Configuration, Catalog Configuration.

Administrators can accept the default settings or optionally change them, as follows:

■ By default, the Use Service Provider Catalog Only setting to No, meaning that users with the authorized roles mentioned earlier (authorized users) can access the forms in the current business unit and its child business units *only*. Similarly, authorized users can add forms to services and delete forms from services in the current business unit and its child business units *only*.

   If you change the Use Service Provider Catalog Only setting to Yes, then authorized users can also view and copy forms in the Service Provider business unit; they cannot perform any other actions on these forms. Similarly, authorized users *cannot* add forms to services or delete forms from services in the Service Provider business unit.

■ By default, the Pass Through Catalog setting is No, meaning that authorized users can access forms in the current business unit and its child business units *only*. Similarly, authorized users can add forms to services and delete forms from services in the current business unit and its child business units *only*.

   If you change the Pass Through Catalog setting to Yes, then authorized users can also view and copy forms in the parent business unit; they cannot perform any other actions on these forms. Similarly, authorized users *cannot* add forms to services or delete forms from services in the Service Provider business unit.

■ If both the Use Service Provider Catalog Only and Pass Through Catalog settings are Yes, then the former setting takes precedence, meaning that authorized users can also view and copy forms in the Service Provider business unit, but not in any parent business unit. Similarly, authorized users *cannot* add forms to services or delete forms from services in the Service Provider business unit.

Note: Users cannot access forms in sibling business units under any circumstances. For details about configuration options, including Use Service Provider Catalog Only and Pass Through Catalog, see the *Implementation Guide*.

Assign a Form to a Different Business Unit

After creating a form, you may decide to copy or move the form to a different business unit, so that its users can use the form also. For example, you may move one or more forms from a folder in Business Unit 1 to another folder in Business Unit 5. Similarly, you may copy a form from Accounting to Finance, from Customer Service to Human Resources, and so forth. If you are an administrator in a parent business unit, then you are automatically an administrator in all of its child business units.

To assign a form to a different business unit, you move or copy (see page 371) the form from the original folder in the original business unit to the destination folder in the new business unit.

Note: You cannot create, copy, move, or delete any forms or elements within the System folder.

How to Localize Forms

After you have finished creating a form in the Form Designer, you can optionally use its Localization Editor to localize the element names and selected attributes in forms. You can do so for one, more, or all of the following languages: Brazilian Portuguese, French, English, German, Italian, Japanese, and Spanish.

While editing the form in the Form Designer, you see localized names and values only when you are using the Localization Editor. However, when a localized form is used in a request, it displays all localized values. If no localized values exist, the form displays the default values for the form name, element names, and the selected attributes.

Note: While editing the form, at any time, you can optionally click Restore to restore the last saved values. This option may be useful if you change one or more values and later realize that you do not want to save the changes.

To localize a form, follow this process:

1. Select the form name in the Component Tree and click the Localize button.

   The Localization Editor dialog appears.

2. Select a language from Language drop-down list.

   The Localization Editor displays the elements in the order that they appear on the form.

3. For each element on the form, such as text fields, radio groups, select groups, and so forth, specify local values for the following attributes. *Only* these attributes can be localized.

   label

   > Specifies the name of the element; this text is used as the name of the field when the user displays the form in a request. For example, you may specify Name, Address, City, or State.

   value

   > Specifies the value of the element; this text is used as the default value for the field when the user displays the form in a request. For example, for an element whose name (label) is Name or Address, you may leave the value blank, because each user must a custom value.

   tooltip

   > Specifies the tooltip text, if applicable. Only certain elements use this attribute.

   patternMessage

   > Applies to text fields and text areas *only*, and only when the HTML attribute (see page 303) named pattern is also used.

   > Specifies the error message to display when a user's input violates the *pattern* attribute.

   > In the value for the pattern attribute, you specify regular expressions to validate numeric and address data (see page 361). Examples include credit card numbers, social security numbers, email addresses, IP addresses, and telephone numbers.

   After you enter a localized value, click Save.

   For example, if you created the form in English and you are localizing it for German, you may use the specifications in the following table to specify the labels for a select group and its options. The first entry is the name of the group and the remaining entries are its options.

Label	Default Value	Local Value
Peripheral Device	Peripheral Device	Peripherie-Geräte
external disk drive	external disk drive	externe Festplatte
speakers	speakers	Lautsprecher
webcam	webcam	Webcam

4. Continue specifying local values for attributes, as explained in the previous steps, until you are finished.

5. Close the Localization Editor by clicking the X at the right top corner.

6. Using the appropriate language settings for your web browser, test the form in a service to verify that the localized values appear correctly for each localized language. If necessary, correct any errors.

Import and Export of Forms

You can use the ixutil command line utility of CA Service Catalog to perform the following tasks:

- Import CA Workflow forms into the Form Designer

- Import and export Form Designer forms by themselves, without related services or service option elements

- Import and export Form Designer forms associated to service option elements

For details about performing these tasks, see the *Reference Guide*.

System Forms

CA Service Catalog provides the following system forms for users to complete as part of their requests: General Information, Request Information. The Request Information form includes Additional Information, and Shipping Information. These forms request user-related information that is typically required to complete a request, such as basic personal data, shipping address, and so forth.

General Information

The General Information form asks the user to provide important details required to complete any check-out process. This form is *not* modifiable and therefore does not appear in the Form Designer. This form appears on the checkout page and other pages of the request.

Request Information Form

By default, the Request Information form asks the user to provide any custom descriptive information about the request. This form is used in all checkout-processes for services in your business unit and any sub-business units.   The default version of the form appears in the Forms, System folder in the Form Designer.

Customization

You can optionally customize the Request Information form by clicking the Customize button in the Component Tree when the form is selected. After confirming that you want to customize the form, you can edit it as you would edit any custom form.

Use caution when modifying this form, because your changes apply to all checkout-process for services in your business unit and all sub-business units, past (even fulfilled requests, no archiving), present, and future.

A tenant can have *one* active custom copy of this form.

Deletion

After customizing this form, you can optionally delete the customized version to revert to the parent form. To delete the customized version, click the Delete button in the Component Tree when the form is open.

The Delete function is disabled for the Service Provider (SP) business unit, because deleting the parent form is prohibited.

How to Hide the Forms

You can optionally hide the Request Information form so that it does not appear to users or administrators during the request life cycle. By default, it does appear.

To hide it, open Service Builder, Configuration, Request Management Configuration, and change the setting named Access Control: Show Request Information to include no roles. As a result, no users or administrators are able to view or edit the Request Information form during request processing.

Information for Upgrades Only

If this installation is your first installation of CA Service Catalog, then this topic does not apply, and you can skip to the next topic. If you have upgraded from a previous release of CA Service Catalog, continue reading this topic.

The Service Builder configuration option named Access Control: Show Request Information replaces *both* of the following Service Builder configuration options from CA Service Catalog r12 and earlier releases:

■ Access Control: Show Shipping Information

■ Access Control: Show Additional Information

In previous releases, these options controlled which roles can view and edit the Shipping Information and Additional Information forms. The Shipping Information and Additional Information forms from previous releases are replaced by the General Information and Request Information forms. Access control for the General Information and Request Information forms is explained in the previous topic.

Restrictions

Keep in mind the following restrictions:

■ Copying and pasting the Request Information form creates a copy that does *not* become the system form.

■ You cannot move the Request Information form out of the Forms, System folder.

■ You cannot alter the Forms, System folder, other than customizing the forms that can be edited.

Pre-Defined JavaScript Functions for System Forms

To better integrate the request information form with the checkout and request edit pages, CA Service Catalog provides the following <u>pre-defined JavaScript functions</u> (see page 342) that you can optionally customize to update fields in the request information form when the general information form is changed. The pre-defined JavaScript functions are as follows:

custom_onPriorityChange(newPriority)

This function is invoked when the priority of the request is changed in the general information form. You can optionally add code here to update the customized request information form to reflect the priority change. By default, this function does nothing.

newPriority is a string that corresponds to the new priority.

custom_onDateRequiredChange(newDateRequired)

This function is invoked when the date required is changed in the general information form. You can optionally add code to this function to update the request information form. By default, this function does nothing.

newDateRequired is a JavaScript date object set to the new required date.

custom_onRequestedForChange(type, id, name, shippingAddress)

This function is invoked when the requested for user or account for a request is changed. You can optionally add code here to update values in your request information form, for example, to change the shipping address fields in the request information form to have the address of the requested for user or account.

By default, this function updates the shipping address section of the request information form with the shipping address of the new user or account.

Important! If the shipping address section of the request information form is changed or updated, this code may need to be updated to reflect those changes.

type is equal to *user* when a user is selected and *account* when an account is selected.

id is the CA Service Catalog ID of the selected user or account.

name is the display name of the selection.

shippingAddress is a JavaScript map with the following keys: address_1, address_2,city, state, zip code, and country. shippingAddress maybe null or empty if the user or account has no address associated with it.

Chapter 9: API Plug-ins

This section contains the following topics:

Overview of API Plug-ins

The network administrator, service delivery manager, and Java programmer work together to write API plug-ins that load data into the following Form Designer fields dynamically:

- Select boxes and options (see page 287)

- Dual lists (see page 286)

- Dynamic tables (see page 290)

The API plug-ins are Java objects that implement one of the following predefined interfaces.

- com.ca.usm.plugins.apis.forms.FDSelectDataProvider: This plug-in applies to select boxes (both single-select and multiselect) and dual list fields.

- com.ca.usm.plugins.apis.forms.FDTableDataProvider: This plug-in applies to dynamic tables.

API plug-ins can query the MDB or another data source and return the number of objects that meet the criteria that you specify. API plug-ins are deployed as jar files in the plug-ins directory of the filestore. API plug-ins run in the same Java Virtual Machine instance as Service View.

You can either write your own API plug-ins or copy and modify the predefined plug-ins to meet your requirements. In both cases, you meet the prerequisites and compile your plug-ins *before* you can use them to populate Form Designer fields dynamically.

When a user completes a form while requesting a service, the report data object populates the specified field with options for the user to select. For example, when a user completes a form to reserve a virtual computer, the report data object populates the list of available computers. In addition, you can write other report data objects to populate related fields, for example, options for RAM and disk space.

How to Write and Use API Plug-ins

To write and use an API plug-in, follow this process:

1. Define the purpose of the plug-in, for example, to populate a select field with meeting rooms that a user can reserve for a specifed time period. Other examples include options for the meeting room, such as projectors, video conferencing units, and microphones.

2. Meet the prequisites. Be able to do the following proficiently:

   ■ Program in Java

   ■ Create forms using the Form Designer

   ■ Create the following fields in Form Designer forms:

     – Single select, multiselect, and dual list fields

     – Dynamic table fields

3. Review the API Plug-in documentation, as follows:

   a. Log in to CA Service Catalog and select Administration, Tools.

   b. In the left menu, select Links.

   c. Scroll down to the Documentation section and click Plug-in Documentation.

   The API documentation is automatically generated Java documentation based on the Java class methods for the plug-ins. You use the interfaces, classes, methods, and so forth to implement your plug-in.

4. Create a Java class for the type of Form Designer field for which the plug-in applies, as follows:

   ■ For single select, multiselect, and dual list fields: Create a Java class that implements the interface named com.ca.usm.plugins.apis.forms.FDSelectDataProvider

   ■ For dynamic table fields: Create a Java class that implements the interface named com.ca.usm.plugins.apis.forms.FDTableDataProvider

   Both interfaces and sample Java classes appear in the API Plug-in documentation. You can use the samples in the API Plug-in documentation as models for your Java classes.

5. (Optional) If you use content configuration forms, retrieve values from fields on these forms and use them as needed.

6. Create a properties file for the plug-in. You can use the samples in the API Plug-in documentation as models for your properties files.

7. If applicable, do one of the following:

   ■ [Configure sorting and pagination parameters for static tables](#) (see page 384)

   ■ [Configure sorting and pagination parameters for dynamic tables](#) (see page 385)

   ■ [Configure Page Size attribute to configure pagination for a select box](#) (see page 308)

8. Create a folder to store the following:

   ■ The properties file

     Important! Store the properties file at the *top* level of the folder. Do *not* store the properties file in a subfolder.

   ■ Any JAR files that provide the classes and supporting libraries

     You can store them in subfolders, if necessary.

9. Do the following to activate the plug-in:

   a. Stop the Windows service for Service View. The service is named CA Service View.

   b. Copy your folder (including all subfolders, if any) to the %USM_HOME%\filestore\plug-in folder.

   c. Restart the Windows service for Service View.

10. Do the following to verify that the plug-in was successfully adopted:

    a. Log in to CA Service Catalog and select Administration, Tools.

    b. In the left menu, select Plug-ins.

    c. Verify that the plug-in is listed and that its details appear properly.

You are ready to test this API for use in a form field.

Configure Pagination Parameters for Select Fields

Your plug-in can return a large amount of data in a select field on a form. In that case, you often specify the page size of the results in the select field. An example is displaying ten results per page in the select field. To accomplish this task, configure the parameters for pagination in the related Java class and object.

Follow these steps:

1. Edit the Java class that implements the interface named com.ca.usm.plugins.apis.forms.FDSelectDataProvider. Implement the following method in FDSelectDataProvider:

   List<FDOption> getOptions(int start, int numToReturn);

   The values of the following parameters determine the paging that users see when they access the select field on the form.

   start

   Specifies the first row to return. This parameter is an integer.

   numToReturn

   Specifies the number of rows to return. This parameter is an integer.

   This method returns a list of FDOption objects. These objects are *both* of the following:

   ■ The key value pair (id and value) in a report data object for the options of the select field.

   ■ Additional data that complement the key value pair. You can optionally display this additional data in other fields (except the select field) on the form.

   The value of the _id attribute of each field must match the one of the keys in the additional data.

2. Also in FDSelectDataProvider, implement the following method:

   int totalCount();

   This method returns the total number of rows that exist.

3. Complete the remaining steps to write and use this API plug-in (see page 382).

Note: For more information and examples, see the API plug-in documentation.

Configure Sorting and Pagination Parameters for Dynamic Tables

If your plug-in returns a large amount of data in a [dynamic table](#) (see page 290) in a form, you often specify the page size of the results. An example is displaying ten results per page on the form. Similarly, you often want to let users sort the results on each page in ascending or descending order. To accomplish both tasks, configure the parameters for sorting and pagination in the related Java class and object.

Follow these steps:

1. Click Service Builder, Form Designer, and open the form you want.

2. Open the table and verify that the attribute named Sortable is set to True.

   If this attribute is enabled (True), then users can click arrows on the table column headers to sort the results in ascending or descending order.

3. Edit the Java class that implements the interface named com.ca.usm.plugins.apis.forms.FDTableDataProvider. Implement the following method in FDTableDataProvider:

   List<<FDTableRow>> getTableRows(int start, int numToReturn, String sortField, boolean sortAscending);

   The values of the following parameters determine the paging and sorting that users see when they access the form. Users determine these values by interacting with the form, for example, by clicking to view the next page.

   start

   > Specifies the first row to return. This parameter is an integer.

   numToReturn

   > Specifies the number of rows to return. This parameter is an integer.

   sortField

   > Specifies the rows to sort. When the value is null, no sorting occurs. This parameter is a string.

   sortAscending

   > Specifies whether to sort the results in ascending or descending order. This parameter is Boolean.

   This method returns FDTableRow objects, which are described in a later step.

4. Also in FDTableDataProvider, implement the following method:

   int totalCount();

   This method returns the total number of rows that exist.

5. Use the FDTableRow object and its methods to return table row data. You can use the following methods:

public void setColumnValue(String columnId, String data)

Specifies the following to set the value for a column:

- ■   String columnid – Specifies "_id" attribute of a component of a table.

- ■   String data – Specifies a value that you can parse and place into the field on the table.

public String getColumnValue(String columnId)

Specifies the matching GET function for the previous SET function (public void setColumnValue).

public Set getColumns()

Returns the set of column IDs stored in this object.

6. Complete the remaining steps to <u>write and use this API plug-in</u> (see page 382).

Note: For more information and examples, see the API plug-in documentation.

Chapter 10: Using CA Service Accounting

This section contains the following topics:

Introduction

CA Service Accounting is a financial application that can be used for controlling IT costs, managing demand through billing and chargeback, allocating costs, budgeting and planning IT Services.

CA Service Accounting enables businesses to allocate their IT costs and create flexible, customized pricing models for their IT services. The software can be used to plan IT budgets and justify budget requests; implement corporate chargeback for IT resources, and bill customers for IT services on a per-use basis.

In addition, CA Service Accounting does the following:

■ Enables usage-based or transaction-based billing by importing external usage through data mediation

■ Adjusts charges based on SLA contract violations, when CA Service Catalog is integrated with CA Business Service Insight

Key features of CA Service Accounting include:

■ A service builder to create flexible, customized options for an organization's IT services

■ A chargeback component to allocate cost back to Business Units, including service level charges and adjustments

■ Scheduled account invoice processing and financial reporting used for budgeting and planning

■ A facility for managing subscriptions, invoices and profiles

CA Service Accounting provides companies with the ability to:

■ Improve Communication and Alignment

A Chargeback implementation triggers dialog over many aspects related to IT Services, such as advantages, costs, and requirements. With a deeper understanding of how IT is being used within an organization, services can be better aligned to meet the business objectives of each Business Unit by correcting or eliminating pervasive inefficiencies.

■ Increase Governance and Financial Transparency

With increased transparency and accountability reported through a successful chargeback strategy, organizations can be more adept in making astute business decisions based on true IT costs.

■ Influence Customer Behavior

Allocating costs to the end-user raises awareness and influences customer behavior. As Business Units become more cost-conscious with their expenditures, inefficient and frivolous behavior can be curtailed, placing better constraints over the cost of IT. The results typically include a more accurate depiction of profitability and increased control over the cost of IT.

- Transition IT toward a Measurement Culture

  By moving Business Units, and the organization as a whole, toward a culture of measurement, increasingly efficient management and continual improvement of overall business processes result.

- Calculate Return on Investment (ROI)

  The ability to identify usage either by consumer or department enables IT to easily calculate the return on investment. The detailed information available in Service Accounting also enables IT to perform cost-benefit analysis.

Service Builder

CA Service Accounting lets IT managers administer and provide access to all their services through the Service Builder menu option. These services can be organized into a folders structure and published to a particular Business Unit or made available to the entire enterprise. A service consists of one or more service option groups, which are used to describe IT services and contain the cost associated with subscribing to the service. This cost can be based on transaction, usage-based billing, rate and more. These service option groups can be comprised of many types of chargeable and non-chargeable elements including application, agreement, and text.

More information:

Chargeback

Enterprises can define their services to allocate costs internally for budgeting and chargeback (CA Service Accounting supports a variety of chargeback methodologies), or to charge customers for them. Charges can be per transaction, or per session, and can vary by resource. The rates used can vary by service package and activity level. When CA Business Service Insight is integrated, the invoices based on real-time usage data can be generated and adjustments can be applied automatically to reflect any SLA violations.

Under most circumstances the cost charged back to a business unit is not disputed, because it can be directly attributed to IT services that have a role in supporting its business operations. The primary focus is on whether or not the services provided and the corresponding costs are at the levels anticipated. In essence, business units want to be certain that they are getting everything that they are paying for. They want to be clear on how to distribute the burden of cost and what measures to take to lower it. Conversely, an organization must verify that all costs are recovered regardless of whether or not resources are fully utilized. In addition, it is important that the process of distributing the cost is easily managed and that the system put into place influences business unit behavior and demands.

There are several approaches to consider when exploring the implementation of a chargeback system.

Note: Before implementing a chargeback system, you understand how to create a service catalog.

More information:

Subscription-Based Pricing

Chargeback can be done without regard to actual consumption, but rather on a fixed basis using a specific unit of measure and unit cost. This is known as subscription-based pricing. This strategy can be implemented in CA Service Accounting by using any chargeable type service option element when defining services for a Business Unit. This element type allows for the specification of a unit cost, billing cycle and quantity. Examples include subscription charges, such as $500 per month or $1000 per laptop.

This method is used when the true cost driver is unknown or perhaps too difficult to ascertain. Since the subscription is based upon a controllable unit of specification, this approach provides the Business Unit with a good deal of control over cost expectation and a clear picture of their IT-related costs, but it fails to meet any objectives pertaining to fairness. Disparate resource use is not accounted for, as the treatment of all subscribers is equal. This decoupling of service cost and usage translates into a system that tends to fail to motivate Business Unit behavior. Consequently, subscription-based pricing does not guarantee the recovery of cost or that profit is generated.

Usage-Based Pricing

Usage-based pricing enables IT to charge back to Business Units cost based proportionately to their consumption of resources. This strategy can be implemented in CA Service Accounting by using the Application type service option element and choosing the Usage Based selection for its Pricing Structure when defining services for a Business Unit. This element type allows for the specification of a unit cost and a metric type.

Once usage based data is imported into the system using data mediation, the metric result can be used to calculate the total cost of that service for an account. In addition to global account charges, these usage-based amounts can be processed on a per user basis.

Implementing usage-based billing involves knowledge and decision at multiple levels within an organization. For example, financial decisions determine how to present and price these services. Also, the process involves IT technical decisions on where and how to collect and process useful usage data, and how to handle exceptional cases like adjustments and promotions.

Usage-based chargeback makes the assumption that the associated resources are shared among several Business Units and their unit cost is fairly static. Fairness is achieved here on some level because consumption is directly linked to the cost incurred. There might be some question as to how charges are influenced by the demands of other Business Units and how this can affect controls on cost.

For instance, when services are set to employ weighted distribution as the means of distributing charges, lowering usage levels of these services does not always guarantee lower cost. In this case, one Business Unit may decide to cut usage back by 5%, thus expecting a relative change in the cost that it is allocated. However, if all other Business Units that contend for the same resources also make a decision to cut back on usage, but at a higher rate, the original Business Unit is forced to pay more than it did previously. Although control can be a problem, this model does have an effect on business unit behavior.

More information:

How to Implement Usage-Based Billing

Tiered-Based Pricing

Tiered-based pricing enables IT to charge back to business units based upon levels of service. A tiered structure can be constructed associating various rates to level ranges. This strategy can be implemented in CA Service Accounting by constructing a tiered type service group to contain these ranges and corresponding rates, and defining services to draw their costing information from them based upon a passed value. You can use this value to look up and select the appropriate tiers.

Consider a tiered service group that contains two columns. In the first column, various ranges can be defined, such as numeric values 1-100, 101-1000, etc. In the second column, associated rates can be defined. You can configure several service group elements to use this tiered service group to determine their cost.

For example, an application service group element can be created to use this tiered service group by selecting Tiered-Based as its Pricing Structure. During the cost calculation for this element, the metric value can be used to select the matching tier(s) and use the corresponding rates. The unit specification at that rate can be derived in a number of ways, such a predefined fixed quantity or the metric value itself. Similarly, agreement service group elements can employ tiered service groups to attribute a cost based on the number of violations.

Depending upon the arrangement, employing a tiered-based structure can be used to enforce equitability and enable business units to make decisions that directly impact their cost.

Combined Approaches

It is more common to have a system that borrows from each of these approaches to create a chargeback procedure that is specifically tailored to the needs of a specific organization. For instance, a particular organization may look to implement a system that influences service objectives through the use of a tiered-based pricing with assurance adjustments on top of a fixed recurring subscription charge. Here the aim is to strengthen the control a Business Unit has over the quality and equitability of a service, while providing for a great deal of stability and predictability of cost.

In addition, a differentiation may exist between services bound to that organization's infrastructure and those that are not. These infrastructure services can be positioned to do chargeback based on a specified allocation method as in a 'per subscription' basis, while chargeback for the other services can be based on actual consumption using a fixed rate.

Invoicing and Financial Reporting

Once services are tied to accounts through a subscription, you can use CA Service Accounting to schedule these accounts for batch invoicing to process any rates, adjustments (general or SLA violation) and payments posted for the current billing period. An invoice group with global settings can be established to bill an associated list of accounts together in one instance at a scheduled future date. This list can be assembled at the time the group is created or determined dynamically during invoice execution. At the specified scheduled time, the invoice engine kicks off and the group is placed into a queue for the accounting service to pick up for processing. Once selected, the engine queries the accounts associated with the invoice group and proceeds to generate an invoice for each of them.

The invoice states and justifies charges to users and business owners. These charges can be rolled up by an enterprise's Business Unit organization and stakeholders, at each level, can be given access to their IT cost information. Once processed, organizations can provide each Business Unit Web access to real time and historical chargeback invoice information, as well as detailed IT usage reports. IT owners can view actual spending versus budget, determine what the actual cost is by business function, and, in general, have much more visibility into IT cost. This strongly facilitates the ability of providing an organization with the necessary data for setting budgets and future business planning, as well as allowing for the continuous improvement and refinement of the cost of IT.

Invoice Groups

An Invoice Group provides the capability for grouping and scheduling accounts for future invoicing. When the invoicing process initiates, each group enters a queue to provide the accounting service the facility to retrieve and process the list of accounts associated to the group. This account list can be assembled and maintained manually or dynamically depending on the group type.

Add a Static Invoice Group

Static Invoice Groups contain an account list that is comprised of accounts that have been manually assigned to it. Accounts assigned to a static invoice group cannot be assigned to any other group. Any accounts assigned to an invoice group remain unless explicitly removed or deleted from the system. The account list for active groups can be further expanded by directly selecting and adding accounts to it.

To add an Invoice Group

1. Click Accounting, Invoices, Groups

   The list of Invoice Groups created for the current Business Unit appear.

   Note: You can change the Business Unit by clicking the Change Business Unit button and selecting one from Search Business Units dialog.

2. Click the Add button.

3. The Add New Invoice Group window appears.

4. Complete the Invoice Group information:

   ■ Business Unit - Indicates the Business Unit to which the group is being added.

   ■ Group Name - The name of the Invoice group you are creating.

   ■ Dynamic - Indicates the group type. Do *not* select it for this procedure, because groups are static by default.

   ■ Comments - Add comments (optional).

5. Complete the Accounting Profile information

   ■ Billing Cycle - Determines the cycle when an account is invoiced. Choices are Daily, Weekly and Monthly.   This field is used in conjunction with Billing Cycle Interval.

   ■ Billing Cycle Interval - Used in conjunction with Billing Cycle, it indicates the number of billing cycles that must elapse before an account is invoiced.

   For example, to specify a quarterly billed account, select a monthly billing cycle, and specify 3 as the billing cycle interval.

   ■ Period Start Date - Indicates when the accounting profile billing period begins. Choices include when the department was created, when the account was opened, or the current date that the accounting profile is created.

   ■ Period End Date – Indicates when the accounting profile billing period ends.

   ■ Days Due Default - Calculates the number of days past the date of the invoice that payment is due.

6. Review the information on the Group Confirmation window. Click Next, if everything is correct or click Back if you wish to make changes.

   The Account List window appears, enabling you to add accounts to your newly created Invoice group.

7. Click Add Accounts and use the shuttle box to move any accounts you wish to include in the account list.   Verify that all selected accounts appear in the Accounts to Save column, and click OK.

8. Click Next and apply the appropriate Scheduler settings giving the billing date and the time zone.

9. Set the status to Active and click Finish.

    The Invoice Groups window appears, reflecting your newly created group.

Add a Dynamic Invoice Group

With Dynamic Invoice Groups, account lists are generated dynamically given a specified criteria stored as a Data Object. As new accounts are added to the system and fall within that criteria, they are included in the account lists associated to that group. In essence, Invoice Groups can now be set to maintain and manage their own account lists.

To add a Dynamic Invoice Group

1. Click Accounting, Invoices, Groups.

   The list of Invoice Groups for the current business unit appears.

   Note: You can change the business unit by clicking the Change Business Unit button and selecting one from Search Business Units dialog.

2. Click the Add button.

3. The Add New Invoice Group window appears.

4. Complete the Invoice Group information:

   ■ Business Unit - Indicates the Business Unit to which the group is being added.

   ■ Group Name - The name of the Invoice group you are creating.

   ■ Dynamic - Indicate the group type. Select this option when you establish dynamic groups.

   ■ Comments - Add comments (optional).

5. Review the Accounting Profile information and make changes to the following fields as necessary.

   ■ Billing Cycle - Determines the cycle when an account is invoiced. Choices are Daily, Weekly and Monthly.   This field is used in conjunction with Billing Cycle Interval

   ■ Billing Cycle Interval - Used in conjunction with Billing Cycle, indicates the number of billing cycles that must elapse before an account is invoiced.

   Example: to specify a quarterly billed account, select a monthly billing cycle, and then specify 3 as the billing cycle interval.

   ■ Period Start Date - Specifies when the accounting profile billing period beginss. Choices include when the department was created, when the account was opened, or the current date that the accounting profile is created.

   ■ Period End Date - Specifies when the accounting profile billing period ends.

   ■ Days Due Default - Calculate the number of days past the date of the invoice that payment is due.

6. Review the information on the Group Confirmation window. Click Next if everything is correct or click Back if you wish to make changes.

7. The Account List window appears empty since there are no associated criteria to assemble this list. Click Define Criteria and use the Account Criteria Builder to create a query to generate a list.

8. After clicking the Create Criteria button, there are two types of criteria specification that can be used in defining the query.

    ■   Save Last Executed Search – saves the last search executed by the Account Criteria Builder's search account mechanism.

    ■   Advanced Custom Search – enables the user to directly specify the where clause in SQL.

9. Once a criteria type is selected, two additional fields are present:

    ■   Name – The name to be associated to this query. (This is stored as a Report Data Object.)

    ■   Comments – Optional field for additional information.

10. Click Save when complete.

11. Click Next.

12. Apply the appropriate Scheduler settings giving the billing date and the time zone.

13. Set the status to Active.

14. Click Finish.

    The Dynamic Invoice Group has been created and is reflected in the Invoice Groups window. Click View Accounting Tasks to view the scheduled task.

Invoice Criteria for Accounts

An account must meet the following criteria before being processed by the invoice engine:

■   Must be included in the Account List associated to an active invoice group.

■   Must have the Billing Date field (in the Accounting profile) specified earlier than the date of the bill run.

■   Must have the status field (in the Accounting profile) specified.

■   Must have the Automatic Invoicing field (in the Accounting profile) set to Yes.

■   Must be subscribed to a service under a valid state (in other words, the subscriptions are not cancelled), if the configuration is set for Allow No Activity Invoices.

Invoice Criteria for Subscriptions

A subscription must meet the following criteria before showing on an invoice:

■ Chargeable items such as one time, daily, weekly, monthly, and installment appear when those time periods for those charges are recognized.

■ An element in the service option groups of a service is not specified to be Excluded From Invoice.

■ Usage and transaction based charges appear when there have been metrics collected to represent use or transactions.

■ Agreement and contract charges as defined through CA Business Service Insight appear on an invoice if it is being charged as a one time, periodic or installment charge.

Specify the Output Location

The default locations where invoices are saved to the hard drive are as follows:

■ Accounts that receive invoices through email have a copy of their invoices stored in %ACC_HOME%\outbox\email.

■ Accounts that receive invoices through regular postal mail have a copy of their invoices stored in %ACC_HOME%\outbox\postal.

■ Accounts that receive invoices through fax have a copy of their invoices stored in %ACC_HOME%\outbox\fax.

■ Accounts that the administrator wants to print internally have a copy of account invoices stored in %ACC_HOME%\outbox\printer.

Note: Specify \\ as a folder separator on Windows systems.

If you do not want to email a business unit an invoice directly, but instead want to print or fax it, you can store the files in a directory on your hard drive. To do this, set the Invoice Method in the accounting profiles of all your customer accounts to Postal by default.

Invoice History

The Invoice history feature enables you to search for and view all invoices for a certain period for any given account for which you have access. Invoice History Records offer a consolidated view of all invoices for a given group. Information displayed includes the Invoice group and the status (successful or unsuccessful). It enables invoices to be viewed in HTML, CSV, or XML format, and provides the facility to email these invoices.

View Invoice Status

As the Invoice Engine processes the invoices, it logs alerts to the CA Service Catalog database. These alerts can be viewed by clicking Messages from the main menu.

Your Alert messages are listed in a table displaying the following parameters:

- Date/Time
- Type
- Source
- Computer
- Message

If a failure occurs during any one of the phases, the Invoice Engine logs a critical severity alert and displays a message.

Proration

Prorating is the process of dividing the cost of a subscription over the billing cycle period of an account. For instance, if a monthly charge is $3000.00, and an account is on a Monthly billing cycle, and an invoice is generated one day into the billing cycle with 30 days left in that month, the charge appears on the invoice with an amount of $100.00 and a description that specifies that the charge is for one day only.

The accounting configuration enables you to specify that both online and batch process invoices are prorated; only online invoices are prorated; only invoices generated during the batch process are prorated; or neither are prorated. By default, both online invoices and batch invoices are configured to prorate charges.

Batch Printing

CA Service Accounting offers a Batch Printing option, allowing you to create print jobs and print invoices individually or as a whole. A print job is a collection of invoices that have been grouped according to the specified credentials. After it has been created, the collection remains unchanged and its invoices may be printed as a whole or individually at any time.

To print in batch

1. Click Accounting, Invoices, Batch Printing.

2. Click Add Print Job.

3. Enter the Name and Description.

4. Select the appropriate Invoice Group.

    An Invoice group is a set of accounts that you can run together in one instance. With the Batch Printing feature, invoice groups serve as an invoice filter. That is, only those invoices that are associated with the accounts in the selected invoice groups are included in this print job.

5. Select Services.

    Select all Services (the default) or further refine your number of invoices by filtering by Service name. Click the search icon to select individual services.

6. Select Billing Cycles.

    By including specific types of billing cycles, the number of invoices in the collection is once again decreased. Note that this billing cycle is for the account, not at the Invoice Group.

7. Click Add.

    Once a print job has been added, the Batch Printing user interface can be used to manage them. Clicking a print job changes its details on the right. You can delete the selected print job or reuse it.

    ■ To delete a print job, select the print job from the list and click Delete.

    ■ To reuse a print job, select the print job from the list and click Reuse. This action discards the old invoices and populates the print job with new invoices. The same print job filtering credentials are used.

    ■ Click Print to print the entire print job.

Note: You can return the invoices in reverse order or filter the invoices by selecting a specific Item Status.

By default, 50 invoices are returned for each page. To adjust this number, specify the desired amount and click Refresh. Clicking an invoice launches it into a new window serving as a print preview tool. Select the invoices to be printed by either checking the check box next to it or using the check all feature. This tool can also check all invoices by item status. By clicking the Reverse Order checkbox, the checked invoices on the current page is printed in ascending order.

After each print item is completed, the status for each item is updated and the print job status appears at the top.

How to Implement Usage-Based Billing

CA Service Accounting provides the ability to set up usage based billing through the data mediation component. To implement usage-based billing while monitoring user dependent metrics, perform the following tasks:

1. Create an application metric, as explained in How to Create an Application Metric (see page 403).

2. Create a service definition, as explained in Create a Service Definition (see page 405).

3. Create the users and account, as explained in Create the Users and Account (see page 406).

4. Create a data mediation profile, as explained in Add a New Profile (see page 432).

5. Define aggregation logic, as explained in Define Aggregation Logic (see page 433).

6. Upload import usage data, as explained in Data Import (see page 435).

7. Aggregate usage data for the fiscal period, as explained in Start Data Aggregation (see page 441).

8. Review the invoice details to see if the usage-based invoice has been processed correctly for the user or account.

   To view the details, including history, for an invoice, navigate the Aggregation Summary pages for the account of interest. Alternatively, click Accounting, Account Management and click the account's action icon named View Invoice History for *account name*.

How to Create an Application Metric

The Data Mediation component uses metric packages to measure usage of an application.

To create an application metric, perform these tasks:

1. Add, edit, or delete event information, as follows:

   a. Select Administration, Data Mediation, Application Metric, Event.

   The Event List window appears.

   b. Click Add to create an event or click Edit to modify an existing event.

   The Event Detail window appears.

   c. Specify all details about the event and click Save.

2. Add, edit, or delete metric information, as follows:

   a. Select Administration, Data Mediation, Application Metric, Metric.

   The Metric List window appears.

   b. Click Add to create a metric or click Edit to modify an existing metric.

   The Metric Detail window appears.

   c. Specify all details about the metric and click Save.

   d. Link or unlink related events, as described in the next step.

3. Link or unlink related events for a metric, as follows:

   Note: Ensure that at least one event is associated with each metric.

   a. Click Associate Event.

   The Metric Name: *metric name* window appears, displaying all events not already associated with the metric.

   b. Optionally, use the Search or Advanced Search option to filter the event list.

   c. Select the events that you want to associate with the metric.

   d. Click Associate Event.

   The selected events are linked and therefore appear in the Associated Events list.

   e. To unlink one or more events that are already associated with a metric, select the events that you want to disassociate and click Disassociate Event.

   The selected events are unlinked and therefore no longer appear in the Associated Events list.

   f. Click Done.

4. Add, edit, or delete application information, as follows:

   a. Select Administration, Data Mediation, Application Metric, Application.

The Application Detail window appears.

b. Click Add to create an application or click Edit to modify an existing application.

The Application Detail window appears.

c. Specify all details about the application and click Save.

d. Link or unlink related metrics, as described in the next step.

5. Link or unlink related metrics for an application, as follows:

Note: Ensure that at least one metric is associated with each application.

a. Click Associate Metric.

The Application Name: *application name* window appears, displaying all metrics not already associated with the application.

b. Optionally, use the Search or Advanced Search option to filter the metric list.

c. Select the metrics that you want to associate with the application.

d. Click Associate Metric.

The selected metrics are linked and therefore appear in the Associated Metrics list.

e. To unlink one or more metrics that are already associated with an application, select the metrics that you want to disassociate and click Disassociate Metric.

The selected metrics are unlinked and therefore no longer appear in the Associated Metrics list.

f. Click Done.

The application metric is created.

Create a Service Definition

To create a service definition

1. Select Service Builder, Service Option Groups from the menu and click the Add button.

2. Complete the Service Option Group Details dialog selecting:

   - Type = Fixed

   - Selection Type = Choose Many

3. Click OK.

4. Verify that the newly saved Service Option Groups appear on the Service Builder, Service Option Groups page. Click the newly created Service Option Group to see the Service Option Elements Definition window.

5. Define the first column to include an Application type element. Click Dbl Click to Add Value and the Service Option Element Definition Dialog pops up. Select:

   - Type = Application

   - Pricing Structure = Usage Based

   - Cost Type = Specify Value

     As a best practice, do not select User Specified for the Cost Type if you are using usage based billing through data mediation logical mapping.   If you select User Specified for the Cost Type, then users can overwrite the default unit cost defined in the original service definition, which you specify in the next parameter. However, data mediation logical subscription is designed to use the default unit cost defined in the service definition.

   - Unit Cost = any numeric value

   - Application = any available application (this example uses Operating System Platform)

   - Metric = any available metric (this example uses Disk Quota Used (KB) Per User).

   - Click Update.

   - Click Save on the Service Option Element Definition.

6. Associate the service option group to a service, as follows:

   a. Click Service Builder, Services.

      The Services page appears.

   b. Click the Add button. In the Service Details pop up, select:

      - Type = Service

      - Selection Type = Can Select Many Times

7. Click Save.

8. Select the Define icon from the Action column. Select the service option group created above to include in the Service and then click Save Selection.

9. Click OK.

You have created the service definition.

More information:

Using Service Builder (see page 139)

Create the Users and Account

To get started implementing usage-based billing, create one or more users and associate them to an account, if you have not already done.

To create the users and account

1. Add the user (see page 81).

2. Add the account (see page 70). While adding the account, asociate the user to it.

You have created the users and account.

Note: If the service activation is to be achieved through request management, then the user is automatically associated to the account.

Generate Invoice

After data is collected for the accounting period, an invoice can be generated to show the charges computed against the accounts usage. If User Usage-Based billing was employed, the charges come from all users who are associated to the invoiced account.

More information:

Invoice Groups (see page 394)
Invoice On-Demand (see page 420)

Financial Reports

Several pre-supplied reports are available to help provide insight and visibility into IT. These reports include:

- Accounting Details – shows the accounting profile details for all accounts given the filter specification.

- Adjustment Details – displays details pertaining to any defined adjustments.

- Invoice Details – shows information related to invoicing such as dates, balance and number of violations.

- Payment Details – provides a listing of payments posted against account balances.

Account Management

- Account Management enables you to administer the accounting activities that can be performed for the account. Using the Account Management functions, you can:

- Add, modify, or delete accounts

- Create, suspend, add notes to or cancel subscriptions

- Maintain invoices

- Modify accounting profiles

- Make payments

- Apply general adjustments

View an Accounting Profile

To view an Accounting Profile

1. Click Accounting, Account Management.

   The Account Management page appears.

2. Select the account.

3. Click Accounting Profile.

   The Accounting Profile window appears and displays related information, such as the account number, billing cycle, and invoice-related specifications.

Note: If you create an account whose status is CLOSED, then you can see the Subscriptions, Invoices, Profile and Accounting Profile tabs. However, you cannot enter a subscription for the account, generate invoices, or edit the Accounting Profile.

Modify an Accounting Profile

The accounting profile contains information associated with an account, such as Account Type, Status, Cycle information, Period Dates and Account Aggregation. Each of these fields tells you specific information about the account and dictates system behaviors. For example, if you close an account, the associated accounting profile handles any residual accounting functions before the status is updated to Closed. The status field moves from "Close Requested" to "Closed" once the outstanding charges and credits are processed for that period.

The accounting profile is automatically generated when an account is created. The default values are derived from the accounting configuration options and can be modified.

Note: For more information on accounting configuration, see the *Implementation Guide*.

To modify an Account Profile

1. Click Accounting, Account Management.

    The Account Management page appears.

2. Select the account.

3. Click the Accounting Profile tab.

    The Accounting Profile page appears.

4. Click Edit to make changes to any of the fields.

Note: If the status of an account is Closed, you cannot enter a subscription for the account, generate invoices, or edit the Accounting Profile.

Aggregation of Accounts

By properly configuring the accounting profiles for a set of accounts, a business unit can designate an account to aggregate the charges and account balances for all other accounts within its scope. To establish this scenario, the accounts selected for this consolidation need to be configured by setting the Aggregate option in the Accounting Profile to Aggregating Account. The aggregating account includes all accounts in the business unit hierarchy, including sibling accounts whose Aggregate option is set to Yes.

In cases where the aggregating account is used to maintain the account balances of all accounts included in its aggregation, the aggregated accounts are typically zero balancing accounts. An account can be configured to operate in this manner by setting its Account Type option to Zero Balance in its Accounting Profile.

Manage Subscriptions

A subscription is the term used in CA Service Accounting to identify the assignment of a service by an administrator to an account. Once the service is activated for the account, any charges stemming from the subscription are calculated for each invoice period for as long as the subscription is active. If a subscription is cancelled, the service is deactivated as soon as all outstanding charges are accounted for and processed. In addition, subscriptions can be suspended for any amount of time, during which no charges are incurred.

You can view and manage your subscriptions by selecting Accounting, Account Management, clicking an account name, and clicking the Subscriptions tab. Although the subscription process is primarily an administrator task, end users are given read access to active subscriptions for accounts to which they belong.

The Configuration, Subscription Management menu option enables Administrators to change the way future subscriptions of an account are handled.

Note: For more information on the subscription management options, see the *Implementation Guide*.

Subscription Types

Subscriptions are possible *only* when CA Service Accounting is installed.

To view the subscriptions for an account, select Home, Administration, Business Units. Next, open the business unit, and open the account. When the account details appear, click the Subscriptions tab to display the Existing Subscriptions window.

Subscriptions Types and How They Are Created

Accounts can have *physical* subscriptions, *data mediation logical* subscriptions, or both types of subscriptions.

- A physical subscription includes one or more service options in a service. A physical subscription results from one of the following:

    - A service explicitly requested by a user is *completed*. Such subscriptions are *request-based* subscriptions. A request that is submitted but is not completed does *not* result in a subscription.

    - The business unit administrator assigns subscriptions to one or more accounts, based on certain criteria, such as business needs. Such subscriptions are *account-based* subscriptions.

- A data mediation logical subscription is virtual; it includes no physical subscription records. Rather, it is a logical mapping and an aggregation mapping. This type results from an administrator using the Data Mediation component to aggregate usage data. In the Invoice component, the usage cost for this subscription is shown when data is aggregated.

    Considering data mediation logical subscriptions is important when you implement usage-based billing (see page 402).

A single account can have one or more subscriptions of either type or both types. However, if an account has both a physical subscription and a data mediation subscription assigned, the physical subscription *always* overrides the data mediation subscription.

On the Existing Subscriptions window, each applicable subscription type appears in the Currently Subscribed column. However, data mediation logical subscriptions appear *only* if no physical subscription exists.

How Subscriptions Are Cancelled

Subscription types are cancelled by the following means:

- A physical subscription (S) is cancelled when the user cancels the subscription for an account.

- A request subscription (R) is cancelled when the user cancels the request for an account.

- A Data Mediation logical subscription (D) is not shown when an administrator unloads or re-aggregates usage data.

Create a Subscription

To create a subscription for an account

1. Select Accounting, Account Management.

2. Click the account name for which you want to add a subscription.

3. Click the Subscriptions tab.

4. Select the service to which you want to subscribe.

5. Click the Subscribe link.

   You are asked if you want to proceed with the subscription changes below.

6. Click Yes.

   The Subscription is added.

Enable Notes in Subscriptions

You can add notes to a subscription if necessary. To do so, make sure you have first enabled this feature in the Subscription Configuration dialog.

To enable notes in subscriptions

1. Click Accounting, Configuration.

   The Accounting Configuration page appears.

2. Click Subscription Configuration.

   The Subscription Configuration options appear.

3. Click the Edit icon.

4. Check "Enable Subscription Notes."

5. Click Update Configuration.

The feature has been enabled and you can add notes to a subscription.

Add Notes to a Subscription

You can add notes to a subscription if necessary. Before doing so, ensure that you have first enabled this feature, as explained in <u>Enable Notes in Subscriptions</u> (see page 411).

To add notes to a subscription

1. Click Accounting, Account Management.

2. Click the account name for which you want to add notes.

3. Click the Subscriptions tab.

4. Click the Existing Subscriptions button.

5. Locate the subscription for which you want to add notes.

6. Click Notes.

   The Notes window appears, enabling you to add notes.

7. Click Save Notes.

The notes are added to the selected subscription for the selected account.

Suspend a Subscription

To suspend a subscription

1. Click Accounting, Account Management.

2. Click the account name for which you want to suspend the subscription.

3. Click the Subscriptions tab.

4. Click the Existing Subscriptions button.

5. Locate the subscription that you want to suspend.

6. Click Suspensions.

   The Add New Suspension dialog window appears.

7. Enter the start and end dates for the suspension or select "Indefinite Suspension."

8. Click Add New.

   The Suspension History window appears again, reflecting your new suspension.

9. Click Close.

The subscription is suspended for the selected account for the period indicated.

Cancel a Subscription

To cancel a subscription

1. Click Accounting, Account Management.

   The Account Management page appears.

2. Click the account name for which you want to cancel a subscription.

3. Click the Subscriptions tab.

   The Subscriptions page appears.

4. Click Existing Subscriptions.

5. Remove the subscription for the service.

6. Click Save.

   A message appears asking you to confirm the subscription changes.

7. Confirm or cancel the changes.

8. Verify that the subscriptions were successfully updated.

Adjustments

Adjustments are credits or debits applied to services, individual charges of a service option group or service option element, and SLA violations. These adjustments can be made by a fixed dollar amount or a percentage. An adjustment can be a general adjustment or an SLA Violation adjustment and can be applied to individual accounts or across an entire business unit. Applying an adjustment to a business unit applies to all accounts within the selected business unit.

How to Apply General Adjustments

You can apply general adjustments either globally to all accounts or individually to a specific account. Perform the procedure that meets your requirements:

■ Apply General Adjustments Globally (see page 414)

■ Apply General Adjustments to an Individual Account (see page 414)

Apply General Adjustments Globally

Applying general adjustments globally may be required when a general adjustment applies to all accounts rather than a small number of accounts. An example may be a global credit to all accounts to cover an accidental overcharge of a specific fee in the previous month's bill.

To apply general adjustments globally

1. Select Accounting, Adjustments.

2. Select General from the side menu.

3. Click the Add General Adjustment button.

   The Add New General Adjustment window appears.

4. Complete the fields, as explained in General Adjustment Properties (see page 414).

   While completing the fields on this window, do the following:

   - For the option named Adjustment For, click the radio button named All Accounts.

   - Optionally, to change the Business Unit, click Search Business Unit icon next to the Business Unit field in the Adjustment definition page.

5. Click OK.

Apply General Adjustments to an Individual Account

At times, only one account may require general adjustments. In such cases, use this procedure to apply the adjustments to the account.

To apply a general adjustment to an individual account

1. Select Accounting, Account Management.

2. Click the Add Adjustments icon (+/-) associated with the account that is to be adjusted.

   The Adjustment Information window appears.

3. Complete the fields, as explained in General Adjustment Properties (see page 414).

4. Click OK.

General Adjustment Properties

The General Adjustment properties appear on the Adjustment Information window. The following fields may require explanation:

Adjustment For

Speciifes the account number of the account being adjusted.

Select Fixed Amount or Percent

Speciifes whether the adjustment is a fixed amount or a percentage.

If you select Percent, provide the following values:

■ Percent - the percentage amount accepts any floating value.

■ Apply adjustment how often - indicates if this is a one time event or if the adjustment percentage is to be applied to each invoice.

■ Apply adjustment to – includes the service, service option column, service option element, all charges, or the total invoice amount.

If you select Fixed Amount, provide the following values:

■ Apply adjustment how often - indicates if this is a one time event, or one of the following: daily, weekly, monthly, quarterly, yearly or every invoice.

■ Amount - indicates the amount of the adjustment.

How to Apply QoS SLA Violation Adjustments

You can apply violation adjustments for QoS SLAs either globally to all accounts or individually to a specific account.

Note: Request service level agreements (SLAs) are a feature of CA Service Catalog, while Quality of Service (QoS) SLAs are available *only* if CA Service Catalog is integrated with CA Business Service Insight. The terms *request SLA* and *QoS SLA* are used when needed in the documentation to distinguish between the two types of SLAs.

Perform the procedure that meets your requirements:

■ Apply SLA Violation Adjustments Globally (see page 416)

■ Apply SLA Violation Adjustments to an Individual Account (see page 416)

To use SLA violation adjustments, you must integrate CA Service Catalog with CA Business Service Insight.

Apply QoS SLA Violation Adjustments Globally

Applying QoS SLA violation adjustments globally may be required when an SLA violation affects all accounts rather than a small number of accounts. An example may be a system-wide outage resulting in denial of services to all accounts for a time period that exceeds the downtime allowed by the accounts' SLA.

To apply QoS SLA violation adjustments globally

1. Select Accounting, Adjustments.

2. Select SLA Violation.

3. Click the Add SLA Violation Adjustment button.

   The Add SLA Violation Adjustment window appears.

4. Complete the fields on this window; these fields are explained in SLA Violation Adjustment Properties (see page 417).

   While completing the fields on this window, do the following:

   ■ For the option named Adjustment For, click the radio button named All Accounts.

   ■ Optionally, to change the Business Unit, click Search Business Unit icon next to the Business Unit field in the Adjustment definition page.

5. Click OK

Apply QoS SLA Violation Adjustments to an Individual Account

At times, only one account may require QoS SLA violation adjustments. In such cases, use this procedure to apply the adjustments to the account.

To apply a QoS SLA violation adjustment to an individual account

1. Select Accounting, Adjustments.

2. Select SLA Violation.

3. Click the Add SLA Violation Adjustment button.

   The Add SLA Violation Adjustment window appears.

4. Complete the fields on this window; these fields are explained in SLA Violation Adjustment Properties (see page 417).

   While completing the fields, do the following:

   ■ For the option named Adjustment For, click the Account Number radio button and specify the account number: you can enter the number directly or click the magnifying glass icon to search for all accounts for the one you want.

   ■ Optionally, to change the Business Unit, click Search Business Unit icon next to the Business Unit field in the Adjustment definition page.

5. Click OK.

QoS SLA Violation Adjustment Properties

You specify the QoS SLA Violation Adjustment properties on the Add SLA Violation Adjustment window. The following properties may require explanation:

Adjustment For

Specifies the account number of the account being adjusted.

Adjustment Type

Specifies whether the adjustment is a fixed amount or a lookup. A lookup refers to a tiered service option group based on the total violations of an SLA package, agreement, or agreement value.

Lookup Service Option Group

Specifies the tiered service option group to use with the lookup adjustment types. The selected item is used to help calculate the adjustment amount.

Per Violation

Specifies the fixed amount to adjust for each violation.

This option applies *only* when the Adjustment Type is Fixed Amount.

Aggregate Violations on Invoice

Specifies that each violation appears as one line item on the invoice.

This option applies *only* when the Adjustment Type is Fixed Amount.

Tier Type

Specifies the type of lookup into the tiered service option group, as follows:

Lookup

Selects the first matching tier.

Variable Lookup

Selects all tiers up to and including the matching tier.

Post a Payment to an Account

Payments can be posted for an account to settle any outstanding balances. An account's balance is maintained in its Accounting Profile.

To post a payment to an account

1. Click Accounting, Account Management.

   The Account Management page appears.

2. Click the Make Payment icon for the account for which you are posting the payment.

3. The Payments List and Payment Details page appears. From this window, you can perform the following actions:

   ■ Add payments to the Payments List by completing the payment details and clicking the Add to List button.

   ■ Modify Payments in the Payments List by selecting the payment, making your modifications and clicking the Update Payment button.

   ■ Delete Payments in the Payments List by selecting the payment and clicking the Delete Payment button.

4. The Business Unit, Account Name, and Account Balance are pre-populated in the Payment Details window.

5. Complete the following Payment Details:

   ■ Payment Date - defaults to today's date.

   ■ Payment Amount - enter the amount of the payment

   ■ Payment Method - select Check, Cash, or Credit Card.

   Note: To configure the selections for payment methods, click Accounting, Configuration, Payment Methods.

   – Payment Response - this field defaults to the property value of the Payment Response of the Accounting Configuration. Default value is Payment received - Thank you.

   – Check information (if payment is by check)

   – Credit Card information (if payment is by credit card)

6. Click Submit Payment.

   The payment is processed.

Manage Invoices

An administrator has several options to manage invoices that have already been generated for an account. The options include editing an invoice, forcing the generation of an invoice, viewing the current standing invoice, and rolling back an invoice.

To manage invoices for an account

1. Click Accounting, Account Management.

   The Account Management page appears.

2. Search for and select an account.

3. Click the Invoices tab.

   The Invoice History page appears.

4. Manage invoices, as follows:

   - View the invoice history

   - Edit an invoice (see page 419)

   - Perform an invoice on-demand (see page 420)

   - View an invoice online (see page 420)

   - Rollback an invoice (see page 420)

You have managed invoices.

Edit an Invoice

An invoice can be edited and line items can be added, modified or removed. These line items include text, charges, adjustments and payments.

To edit an invoice

1. Click the corresponding edit icon to edit an invoice.

   - To add a line item, click the Add Item link under the desired section and complete the associated dialog.

   - To edit existing line items, click the Edit link of the item that needs to be altered.

   - Click the Delete link for any items that you want to remove entirely.

2. Click the Save Invoice link to save all changes made and return to the account's invoice history page.

Invoice On-Demand

On-demand invoicing enables an administrator to bypass the scheduled invoicing time and instantaneously produce the current standing invoice for an account.

Performing an invoice on-demand produces the same results against the account as if it had been processed as a scheduled bill run. A new invoice is generated for the current billing period and the accounting profile is updated.

View an Invoice Online

Online invoicing enables you to view your charges in real-time with a web browser. An online invoice displays charges (credits or debits) that an account has incurred since the ending of its last billing period, up to and including the current date. All the charges are computed in real-time; since these values can change, the invoice does not get persisted until the end of the account's billing cycle when the invoicing process runs.

Similar to on-demand invoicing, online invoices do not have to go through the same account query and verification: an invoice is requested to be viewed for that account regardless of the account status, billing cycle, or automatic invoicing. However, the criteria still apply for choosing or selecting which subscriptions show up on the invoice as charges, as with the invoice process.

To enable or disable prorating charges when viewing your invoice online, set the Pro Rate Online to Yes in the Invoice Engine Configuration of Accounting Configuration.

Rollback Invoice

Rollback invoicing enables you to invoice accounts from a previously billed period. This sequential rollback of invoices provides the ability to reprocess an invoice to reflect any changes made to charges, debits or credits. To rollback to a previous invoice period, select the invoices to rollback from the account's Invoice History window and click Rollback. The period date, bill date and account balance for the account's accounting profile is updated up to the period invoices were rolled back. These Invoices and their associated transactions are permanently deleted from the system once the rollback is complete. Invoice On-demand can be used to regenerate any or all the invoices whenever necessary.

Budgets and Plans

Implementing a successful budgeting process is a primary key to achieve organizational business objectives. Without an effective process in place, performance evaluation is virtually impossible. Budgets can help IT managers evaluate the performance of a business unit though comparison analysis of actual expenditures against budgeted amounts. These findings can aid in justifying IT costs or identifying why expectations have not been met. In addition, this information can be used to establish new goals and discovering the means to reach them through planning.

Through the Budget and Planning module, a financial manager can set up period budgets for business units and their services. At the end of each fiscal period, a variance report can be generated to evaluate the differences between actual and budgeted costs. This information may be then used to help identify the root cause for areas of inefficiency.

Fiscal Periods

Fiscal Periods can be user defined for any time period to represent an accounting period. CA Service Accounting allows for the definition of monthly, quarterly or yearly fiscal periods. Once defined, these periods are used in Budget and Planning worksheets when defining budgets or setting costs for services, and for data mediation when selecting a period for aggregation.

Note: For more information on fiscal periods, see the *Implementation Guide*.

Sets

Sets enable you to define related financial and quantitative values. For example, you can use sets to represent various types of budgets, especially when you are initially creating these budgets. Sets also provide the facility to link costs to services. Sets can be defined to represent a unit cost or total cost of a service. These types of sets are typically associated with a service during service definition when you employ a Cost Type of Allocate Cost or Standard Cost.

To create a new set

1. Click Accounting, Budget and Planning and click Sets.

2. Click the Add Set button.

3. Complete the following information:

    ■   Name – name used to identify the set.

    ■   Description – optional text area to provide a description.

    ■   Status – specifies whether the associated worksheet values can be modified. Choices include Locked or Unlocked. Once a set is locked, values are read-only in the worksheets until the Status is changed to Unlocked.

    ■   Source – provides for further classification of the associated values. Choices include:

        –   Allocate Cost – used when defining the total cost of a chargeable service. Once this type of set is bound to a service, an associated Allocation Method can be selected to employ a means of distributing the cost.

        –   Standard Cost – used when defining the total unit cost of a chargeable service. Once this type of set is bound to a service, a predetermined service cost can be set for each period.

        –   Actual Units – used to attribute unit quantities to services.

4. Click Add New Set.

More information:

Using Service Builder (see page 139)

Use Worksheets

Worksheets are used to capture budget and cost totals for services. A Service Worksheet can be used to set budgets, unit cost or total cost for services across any number of fiscal periods. Costs can also be pooled and the Service Worksheet can be used to set service costs by tracing pooled activities to the services that consume them. A Pool Worksheet is used to establish these pooled costs. A cost pool is typically comprised of one or more contributing cost elements whose sum makes up the total cost of the pooled activity. A percentage of this total can then be allocated among the services in the Service Worksheet.

To use Worksheets

1. Click Accounting, Budget and Planning, Worksheets.

   The Worksheet Options page for the current business unit appears.

2. Click Change Business Unit to select a different business unit, if necessary.

3. Complete the Worksheet Options:

   ■ Type – either Service Worksheet or Pool Worksheet.

   ■ Period – period type of the fiscal period.

   ■ From – fiscal period start date.

   ■ To – fiscal period end date.

   ■ Sets – list of available Sets.

   ■ Pools – list of available Cost Pools

4. Click View Worksheet.

The services presented in the worksheet are those that contain any chargeable service option group elements whose Budget option is selected. For the elements that employ a Cost Type of either Allocate Cost or Standard cost, the values entered in the worksheet are used to determine the cost of the service element.

If you use the Allocate Cost setting, its value represents the total cost of the service element to be allocated back to Business Unit accounts. The distribution of that cost is determined by the associated Allocation Method specification. For instance, if the method is set to Distribute by Subscription, then the total cost of the chargeable element is divided by the number of subscriptions to it.

The Standard Cost option, however, is used to define a predetermined unit cost for the service option group element. This cost can be used to assign a rate to a service that may vary from period to period, or as the expected costs defined in a budget. For instance, there may be a need to predefine a set of unit costs to be used for a particular period, but then change to another at a particular junction. Worksheets are the tools that enable these service rates to change over time. Similarly, worksheets allow for the definition of budgets for service over a specified set of fiscal periods. The values defined in this instance are not used in the cost calculation of a service option group element, but are used instead in reporting variance between budgeted and actual amounts.

More information:

Activity Based Costing (see page 426)
Using Service Builder (see page 139)

Create a Cost Element

A Cost Element is a resource used to subdivide costs corresponding to the consumption of a particular service. It is the amount paid for a resource consumed by an activity and included in a cost pool.

To create a Cost Element

1. Click Accounting, Budget and Planning, Cost Elements.

   The Cost Elements page appears.

2. Click the Add Cost Element button.

   The Definition dialog for cost elements appears.

3. Complete the fields for this cost element.

4. Click the Update button.

You have created a cost element.

Create a Cost Pool

A Cost Pool is the grouping of all cost elements associated with an activity carried out by an entity. It can be used to identify the cost of any main activities in providing a service.

To create a cost pool

1. Click   Accounting, Budget and Planning and click Cost Pools.

    The cost pools page appears.

2. Click the Add Cost Pool Button.

    The Add New Pool dialog appears.

3. Complete the fields on the dialog:

    ■   Name – name to identify the cost pool

    ■   Type – specify how Cost Pool totals applies to service costs.

        –   Fixed – fixed amounts are used.

        –   Percentage – a percentage of the Pool total is used.

4. Click Add New Pool button.

Assign Cost Elements to Cost Pools

Once the main cost pools have been defined, a total cost of each cost pool can be calculated. First, the activities related to each cost pool are identified and assigned to the appropriate cost pool. To properly trace the expenses to each cost pool, cost drivers have to be identified for each cost element.

To assign a cost element to a cost pool

1. Go to Accounting, Budget and Planning, Cost Elements.

2. Select the Cost Element(s) you wish to assign to a cost pool.

3. Click the Assign to Cost Pools button.

4. Assign pools with cost elements by selecting the corresponding checkboxes.

5. Click the Save Assignments button.

Activity Based Costing

Activity Based Costing (ABC) is a process of assigning costs to services and is typically used as a tool for planning. The method aims initially to identify and categorize the main operating activities within an organization. Once defined, the costs attributable to these activities are assigned to cost pools, which are then allocated back to services based upon how much the activity is consumed by the service. The steps that follow outline the ABC process:

- Typically, an organization can be broken down into a set of succinct activities that accomplish business processes. Cost pools are created to assemble all cost elements related to each of these activities. Refer to Cost Pool section on how to create a cost pool.

- Once activities have been identified, the principle causes of the costs for each activity needs to be identified. These expenses are categorized by establishing cost elements.

- After adding the cost elements, they must be related to their corresponding cost pools through cost pool assignment.

- The Pool Worksheet is used to apply values to the cost pools based on the expense attributed to each associated cost element. Go to Accounting, Budget and Planning, Worksheets and use the Pool Worksheet option to apply these values.

- The Service Worksheet is used as activities are traced to services. The associated pooled costs are allocated to services based on their level of consumption of activities. The Service Worksheet is used to capture these amounts, which can be fixed or a percentage of the pool total.

- Only services that contain chargeable service option group elements are eligible to appear on the Service Worksheet. This is accomplished by checking the Budget properly in the service option group element definition dialog when defining a service. In addition, to having the worksheet values applied in invoicing calculations, a cost type of Allocate Cost or Standard Cost must be selected for the service element.

- Invoices generated for accounts subscribed to these services show an expandable breakdown of services by cost pools and cost elements.

More information:

Chapter 11: Data Mediation

This section contains the following topics:

Data Mediation

Data Mediation lets you import usage event data from various external sources. Data Mediation uses the process of ETL (Extraction, Transform, and Load). You can use this process to transform operational data into event data to support billing, and reporting. After the data has been imported, the data can be manipulated and used by other CA Service Catalog components. Data Mediation is useful when batch operational data from a source other than CA Service Catalog is available and the requirements allow for historical rather than real-time data.

Data Mediation supports the following types of data feeds:

- Delimiter Separated File

- Fixed Length File

- Database Import (includes Database Table Import and Advanced Database Import)

When the external usage data is imported through Data Mediation as File Type data feeds, it is transformed into a flat file and is uploaded to the \datamediation subdirectory of the filestore location. By default, the filestore location is the %USM_HOME%\FileStore\ directory of the Service View computer that you are using. To verify the location of the imported flat files so that you can view them, verify the filestore location by selecting Administration, Configuration, File Store Information. Note the value of the filestore location and use it to find and view the flat files in the datamediation subdirectory; the value may be a local directory, or it may be a directory on another Service View computer.

Important! The folder name *FileStore* is case sensitive. Therefore, use the correct case in path names and all other programmatic references.

Note: For details about the filestore, see the *Implementation Guide*.

Depending on the completeness of the usage data, the data is loaded into the database into a *reference table*, a *temporary data table* or a Data Mediation *event table*. Mapping between external data feed fields to database field types must be defined prior to importing data. Field definitions also contain the Driver Validation Rules which are enforced upon data import and filter irrelevant data.

Data Aggregation

Data Aggregation is the gathering and expressing of data in a summary form. When Data Aggregation is launched, data from the reference tables or temporary data tables are normalized into a set of Data Mediation event tables.

Aggregation processes one event at a time. Normalization or initial aggregation is customized using a SQL query and is specified in the Data Mediation profile. Data stored in Data Mediation event tables represents complete usage event data. All data from Data Mediation event tables is aggregated and the results are loaded into the set of metric-based result tables. These metric requests are processed for the CA Service Accounting rating engine. Budgeting information and billing transactions are created based on the resulting data contained within the metric result tables.

Imported usage data by defined profiles is processed for CA Service Accounting through Data Aggregation. The usage data is imported using the Metric Data Profile. Before usage data is aggregated, the data must be normalized into a common format. The common format consists of fields known as Server Mandatory fields. There are five Default Server Mandatory Fields which are required in every record to qualify as complete event usage data.

CA Service Accounting creates billing transactions based on the event results. The data imported through Data Mediation is aggregated by fiscal period. The data can be aggregated for a defined fiscal period or for all fiscal periods. Aggregation produces results by server mandatory fields, for example, by account, service offering, metric, and fiscal period.

Data Summary

While designing Data Mediation, it is important to determine where to summarize data. For example, if your business objective is to measure average CPU usage over a month for reporting, you have various options. If the data source system has running information and can determine the average over the period, then use a single, summarized data point rather than a data series of five second samplings over the same time period. This data point serves to efficiently use computer and human resources (CPU, bandwidth, maintenance). The best practice is to summarize information early in the process.

Define Data Fields

Extraneous fields from an external data feed need to be defined to map the external data feed field to a database field type. You can also define field definitions when creating a profile. When creating a profile, you use field definitions to define the structure of the external data.

When usage data is imported, it is uploaded into the database. During data import, data validation checks are performed. Invalid records are not uploaded into the database. Before data is imported into the database, you can apply rules to filter out erroneous or irrelevant data. Field definitions hold the definition of these validation rules, specifying a validation rule on each column of data imported.

To define data fields for Data Mediation

1. Select Administration, Data Mediation.

   The All Fields list of existing fields displays.

2. Click Add to define a new field.

   The Define a Field window displays.

3. Specify the following information:

   Display name

   Specifies the field name that appears on the GUI to end users.

   Field name

   Specifies the database field name.

   Data type and size

   Specifies the database field type and size.

Mandatory Types

- None - indicates that neither the server nor the client is mandatory.

- Server Mandatory - indicates that the field is required for data mediation aggregation to occur. The absence of these fields results in a profile error.

- Client Mandatory - includes the field in profile definitions. If the field is not included, a warning message appears when the profile is created.

- Both - indicates that both the server and client are mandatory.

- **Default:** Server Mandatory

Check type

Specifies the check type, if applicable.

- **Default:** No check

4. (Optional) Specify a driver validation rule:

   Not Empty

   Specifies that the field value must not be empty.

   Range

   Specifies that the field value falls within the specified range.

   Lookup

   Specifies that a field value exists in a database table field.

   Lookup Replace

   Specifies a replacement value, if the value in the field exists in the specified database table field.

   These validation rules are enforced when usage data is imported. If the driver validation rules are violated upon import, a kickoff report is generated. The report is available through the Data Management user interface.

5. Click Submit.

   The data fields are defined.

Default Server Mandatory Fields

Usage data has to be normalized into a common format understood by data mediation before it is aggregated. The aggregation results in an error if the following server mandatory fields are not included:

CA Service Account Number

Associates a data record to a CA Service Accounting account ID.

Event ID

Associates a data record to an event (Event ID).

Event Time

Associates a data record to a fiscal period (Time Stamp).

Metric Value

Associates a data record with a metric value.

Service Code

Associates a data record with a service offering (Service Offering ID).

Note: Formatting data to include all server mandatory fields prior to data import is unnecessary. Data Mediation features complex data mapping mechanisms, such as validation and normalization, which help the administrator include server mandatory fields.

How to Implement Data Mediation

The following steps describe the overall process for implementing Data Mediation:

1. Create profiles for each data set that is to be mediated and then aggregated.

2. Import the data (files, external DB) for the associated profile.

   Note: Table names and columns names in the data being imported must be in English.

3. Ensure your fiscal period has been defined to fit your business model.

4. Aggregate data (one or more times) so that the metric result and transactions are created. Repeat the aggregation process if any new data is imported or changed for that fiscal period.

5. Generate the invoices to reflect the data being aggregated, after all the data is imported and aggregated.

Profile Management

Profile management is a definition of the data feed of the external data, providing powerful features to manipulate and normalize usage event data. The profile also contains the initial aggregation logic in the form of an SQL statement. This initial aggregation logic is specified during profile definition.

If necessary, usage event data can be normalized or aggregated, preparing it for the Data Mediation's aggregation and cost allocation processes through a SQL query or procedure defined in the profile. This profile feature helps normalize the data to a common format understood by Data Mediation.

Create a Profile

You can create a profile to define the data feed of the external data.

To create a profile

1. Click Administration, Data Mediation, Profile Management.

    The Profile Management list appears.

2. Click Add New Profile.

    The Profile Definition page appears.

3. Select the Data Type:

    Metric Data

    > Specifies the usage event data for the cost allocation process. This type of profile loads data into temporary or event tables based on the completeness of the usage event data.

    Reference Lookup Data

    > Data contains reference data for lookup purposes. Data is loaded into the reference tables for data mediation.

4. Select the Import Format. Choose one of the following formats:

    ■   Delimiter Separated File

    ■   Fixed Length File

    ■   Database Table Import

    ■   Advanced Database Import

5. Click the Define Fields button.

    A profile template window appears for field definitions.

6. Enter a unique name for the profile you are creating. Define the fields for the profile.

7. Click Save.

    The profile is created.

Define Aggregation Logic

You can define the initial aggregation logic in the form of an SQL statement.

To define aggregation logic

1. Click Administration, Data Mediation, Profile Management.

   The Profile Management list of profiles appears.

   If the target table column is empty and the configuration status is "pending," the initial aggregation logic for the profile is not yet defined. The presence of a target table entry and a configuration status of "Defined" indicate that the profile has been previously defined.

   Note: The Target Table column is visible when the Show more columns double arrow icon is clicked.

2. Click the Define Aggregation Logic icon or the name field for the profile whose aggregation logic is to be defined.

3. Drag-and-drop each field from the data source into the target table field to define the logic to be used during aggregation.

4. Click Generate Query to generate the SQL Expression.

   Note: You can define the initial aggregation logic by using a SQL statement or procedure. Specify the initial aggregation logic to prepare the imported event usage data to include all of the server mandatory fields for the aggregation and cost allocation process. For example, suppose you have usage data for each host in temporary tables for data mediation and want to map the hosts to accounts using the data mediation reference tables. In that case, perform a join of these tables in the SQL procedure to prepare the event table.

5. Click Save.

   The aggregation logic is defined.

Define Multiple Aggregation

Data imported from an external source may need to be aggregated in multiple ways. To fulfill this requirement, you can define multiple profiles based on the same data set. This definition lets you use one set of external source data and aggregate it in different ways to meet your business requirements.

To define multiple aggregation logic on a profile

1. Go to Administration, Data Mediation, Profile Management.

2. Click Copy for the profile for which you want to define a new profile and different aggregation logic.

   A copy of the profile using the same name is created using the same source. The SQL Expression is not copied.

3. Click the new profile and define the alternate aggregation logic for this data source. Click Save.

   A different target table entry is created.

   When data aggregation is called, the SQL query or procedure runs, resulting in two separate event tables. If the data in both tables is normalized properly, both data sets are aggregated successfully.

   Note: Initial aggregation logic for a profile is run without regard to sequential order.

Edit a Profile

You can edit a profile.

Note: You cannot edit profiles after they have been aggregated. If the Import Status is "Processed," you cannot edit the profile.

To edit a profile

1. Click Administration, Data Mediation, Profile Management.

2. Click the Edit icon for the profile you want to edit.

   The Profile Definition window appears and you can edit the fields.

3. Click Save.

   The profile changes are saved.

Delete a Profile

You can delete a profile.

Note: You cannot delete data profiles after they have been aggregated.

To delete a profile

1. Click Administration, Data Mediation, Profile Management.

2. Click the Delete Profile icon Delete icon for the profile you want to delete.

   A warning appears stating that any imported data related to this profile is to be deleted.

3. Click OK.

   The profile is deleted.

Data Import

After you create a data mediation profile and define the data feed and structure of the external data, you can import the data. You can import database profiles as needed and schedule them to occur at a later time.

You can import data from a file or from an Ingres, SQL Server, or Oracle database table. Data is loaded into database tables when imported.

The database tables created by Data Mediation include the following types:

Reference tables

Contains reference data to be used as lookup tables (usm_mr_iref).

Temporary tables

Contains partial usage event data, meaning that all server mandatory fields are not present, implying that normalization may be necessary (usm_mr_itemp).

Event tables

Contains complete usage event data (usm_mr_ievent).

Metric result tables

Contains aggregated event data (usm_mr_iresult).

Data Profiles

Every data profile has one data source. The data profile contains a definition of the data feed of the external data. The profile also contains the initial aggregation logic in the form of a SQL statement. This initial aggregation logic is specified during profile definition. A data source may consist of an ASCII file (fixed-length, or delimited) or a database query. For database queries, multiple physical database resources (tables, databases) can be combined into a single result set and is considered a single data source.

One set of fields exist per profile. Consider the size of the returned dataset to determine how to deal with portions of the total data so that you can perform loads, unloads, and aggregations in smaller sets.

Unique initial aggregation logic (normalizing and aggregating raw data) indicates a unique profile. However, given the same data source and fields you can copy an existing profile to create a profile without having to redefine the source information. An example of an appropriate time to use this technique is if given a single data source, aggregation logic varies based on some field value.

The following are the various types of profiles for Data Mediation:

- Metric Data—Usage event data for the cost allocation process of CA Service Accounting. This type of profile loads data into   temporary or event tables based on the completeness of the usage event data.

- Reference Lookup Data—Data contains reference data for lookup purposes. Data is loaded into reference tables.

If necessary, usage event data can be normalized or aggregated, preparing it for the aggregation process and cost allocation process using an SQL query or procedure defined in the profile.

To import data using Data Mediation

1. Go to Administration, Data Mediation, Data Management.

2. Click Import Data.

3. Select the appropriate Data Type of the profile.

4. Select the import format: Database Table Import or Advanced Database Import.

5. Select the appropriate import profile for the data to be imported.

   You can specify a filter and description.

6. Click Import Now.

Import Data

You can import data by using Data Mediation.

To import data using Data Mediation

1. Click Administration, Data Mediation, Data Management.

2. Click Import Data.

3. Select the appropriate Data Type of the profile.

4. Select the import format: Database Table Import or Advanced Database Import.

5. Select the appropriate import profile for the data to be imported.

   Note: You can specify a filter and description.

6. Click Import Now.

   The data is imported.

Schedule Data Imports

To further automate the process of importing usage data, the CA Service Catalog scheduler can schedule tasks to import data on a time interval defined. The scheduler can import data on a daily, weekly, monthly, or yearly basis.

To schedule a data import

1. Click Administration, Data Mediation, Data Management.

2. Click Import Data.

   The Import Definition page appears.

3. Select the appropriate Data Type of the profile:

   ■ Metric Data

   ■ Reference Lookup Table

4. Select the Import Format:

   ■ Database Table Import

   ■ Advanced database Import

5. Select the profile you want to import.

6. (Optional) Enter a Description of the profile.

7. Click Schedule Import.

   The scheduling information is pre-populated. The administrator only needs to schedule the task by entering an optional Comment and the required fields.

8. Completed the required fields:

    Valid From

    Specifies that the scheduled task is valid from this date.

    Valid Until

    Specifies that the scheduled task is valid until this date.

    Recur

    Specifies that the intervals at which to repeat this task. The value specified in this field causes the Rule field to change.

    Rule

    Specifies the time and how often to repeat the scheduled task.

    Note: Do not modify fields other than Comment and the required fields.

    Click OK.

    Note: To edit a scheduled database import, go through the Scheduler user interface and click View Scheduled Tasks.

    The database profiles are scheduled for import.

Data Management

Data Management lets administrators manage data sets imported into the CA Service Catalog by using the following functions:

Unload Checked

Unloads (removes) the data from the data mediation tables. Associated charges are removed the next time aggregation occurs.

Note: You do not need to roll back the invoices that used the data provided by this data set before unloading and performing the re-aggregation.

Re-import Checked

Reloads imported data sets and effectively resets the aggregation flag of the imported data set to "not aggregated." Associated charges are recalculated the next time aggregation occurs.

Note: For information regarding Importing Data and Scheduling Data Imports, see the *Reference Guide*.

Archive Checked

Archives the selected data set. The status of the data set is changed to Archived.

Note: Archived data sets cannot be unloaded or re-imported after archiving. We recommend that you do not use archiving unless absolutely required.

Note: Aggregation results for a period appear on the Aggregation Status page, although data for that period has been unloaded and re-aggregation performed.

Custom Fields

You can use custom fields to facilitate aggregation logic. For example, business rules may indicate that the processing of rows in a data set varies based on some external value.

You can also use custom fields to maintain legacy information as part of a user interface customization. For example, a charge may have a value in an external system (such as, accounts receivable) that is used to perform cross referencing.

Reference Data

Reference data is often required to facilitate the transformation process. An example is the need to translate account codes from a legacy system to match accounts in CA Service Catalog. In such a case, you can match accounts in the following ways:

- You can customize the user interface to add a field for the legacy account code in CA Service Catalog. This option is best to use when the mapping information is static and can be maintained through data entry.

- You can use a reference table. This option may be required if the mapping data is dynamic and maintained outside of CA Service Catalog.

Data Import Frequency

You must decide the frequency of data imports. To determine the frequency you must consider your reporting requirements.

For example, consider the following:

- If an invoice, an SLA report, a metric report, or a custom report is run daily, then the Data Mediation feed frequency must also be daily.

- If the data is required for a monthly bill run, then a monthly feed may be appropriate.

■ The volume of data being imported. For example, if millions of records are to be loaded monthly, it may make sense to import subsets daily to spread the processing and minimize the end-of-month window. Another benefit of this granular approach is to avoid having to unload and reload larger data sets if changes are necessary. The size of the data set does not affect the commit size in the database tables. The best practice is to perform commits in small batches, rather than after the entire data set. This approach minimizes lock contention and log file growth.

How to Aggregate Data

Every time an aggregation is run for a defined fiscal period, the transactions created in the previous aggregation are deleted and new transactions are created to reflect the data being imported. For example, if you generate a CA Service Accounting invoice after an aggregation and then re-aggregate the data for the same fiscal period, the previously generated invoice has zero charges. This result is due to the transactions being rolled back on re-aggregation. New transactions are created as part of the re-aggregation. Therefore, invoices need to be regenerated to capture these new transactions and show the associated charges.

You can use Invoice groups and then generate your invoices in batch mode. You can also bill for two different data sets that are imported at two different times. You can do an aggregation after the first one is imported and then aggregate again after the second data imported. Only generate invoices after you verify that all the data for that fiscal period is valid.

To aggregate data, follow these steps:

1. Generate usm_mr_itemp_XXXXXX data from the external files and databases.

2. Generate usm_mr_ievent_XXXXXX data from the usm_mr_itemp_XXXXXX data based on the SQL aggregation specified in the Data Mediation profile. The XXXXXX values are sequential numbers generated by the system.

3. Generate usm_mr_iresult_YYYY data from the usm_mr_ievent_XXXXXX table based on the built-in aggregation logic. The YYYY is the internal metric ID for the event specified by the aggregation.

4. Generate transactions for the accounts based on the usm_mr_iresult_YYYY data created in Step 3.

You can run the aggregation process any number of times for a fiscal period to reflect the latest data that is imported into the system.

Start Data Aggregation

You can aggregate data.

To aggregate data

1. Select Administration, Data Mediation.

   The All Fields list of existing fields displays.

2. Select Aggregation Status.

3. Complete the window options:

   Use selected period for aggregation

   Aggregates data contained within a specific fiscal period (suggested). Only data falling within the specified date range is aggregated. Not selecting this option aggregates data for all fiscal periods.

   Advanced Option

   Specifies multiple fiscal periods.

   Click Start Aggregation.

4. (Optional) Click Refresh to check the Aggregation Status during data aggregation.

   The data is aggregated.

Aggregation Status

The possible data aggregation status values are as follows:

- Load files only—New data has been imported since the last aggregation. Aggregation is not currently occurring.

  Numeric code: 0

- Importing—Data is being imported and initial aggregation logic in profiles is being executed. In this phase, any data in reference tables or temporary data tables is being placed into Data Mediation event tables.

  Numeric code: 1

- Aggregating—Aggregation is being performed. DC Importer is performing aggregation. In this phase, the data in Data Mediation event tables is being stored into the metric results tables.

  Numeric code: 6

- Profile Error—An error occurred while executing initial aggregation logic contained in profiles. Aggregation is stopped. Go to Profile Management to locate profiles that are in error. Profiles in error are denoted with Error in the Import Status column.

  Numeric code: 8

- Profiles Pending—A profile has not been defined. Aggregation is stopped. Profiles containing usage event data must be defined.

  Numeric code: 7

- Waiting for Billing to Finish—The CA Service Accounting rating engine is creating billing transactions.

  Numeric code: 5

- Aggregation Error—A severe error occurred during aggregation.

  Numeric code: 4

- Aggregation Complete—Aggregation completed successfully.

  Numeric code: 3

How to Generate Invoices

You can use data aggregation to generate invoices in CA Service Accounting. You can run this aggregation process any number of times for a fiscal period to reflect the latest data that is imported into the system. Every time an aggregation is run for a defined fiscal period, the transactions created in the previous aggregation are deleted and new transactions are created to reflect the data being imported. If you generate an invoice after an aggregation, and again aggregate the data for the same fiscal period, the previously generated invoice has zero charges. These zero charges are due to the transactions being rolled back on re-aggregation.

New transactions are created as part of the re-aggregation. You regenerate invoices to capture the new transactions. You can use invoice groups and create invoices in batch mode. If two different data sets are involved at two different times, you can aggregate when the first data set is imported and again aggregate after the second data set is imported. You then generate invoices after all the data for that fiscal period has been received and verified.

To generate invoices using Data Mediation, do the following:

1. Create profiles for each data set that has to be mediated and then aggregated.

2. Import the data (files and external database) for the associated profile.

3. Define fiscal periods to fit the business model.

4. Aggregate data (one or more times) so that the metric result tables and the CA Service Accounting transactions are created. Repeat the aggregation process if any new data is imported or changed for that fiscal period.

5. After all the data is received and is aggregated, generate the invoices that reflect the aggregated data.

Data Mediation Reports

There are several Data Mediation report data views available including:

Data Mediation - Profile List

Displays a list of Data Mediation profiles, including the following associated parameters:

– Index ID

– Profile Name

– Status

– Temp Table

– Event Table

– Data Import Type

– Profile Owner

– Description

– ID

Data Mediation - Rating Engine Queue Items

Displays rating engine queue items by the specified runtime variables. Enter the start and end date and times. The report displays the Data Mediation Rating Engine Queue items, along with the following associated fields:

– Index

– Group ID

– Queue Item ID

– Created Time

– Status

– Start Time

– Finish Time

– Metric Result ID

– Metric Result Table Name

Repository Agent

The CA Service Repository Agent (also known as Data Mediation Data Repository Agent) automates the process of importing usage data stored in Delimiter Separated File or Fixed Length File format. The repository agent is installed as a service named CA Service Repository Agent during the installation of Service View.

The Repository Agent imports usage data files into the database in the following ways:

■ The agent can be configured to read files from an FTP server. The files are automatically copied from the FTP server locally and processed based on associated Data Mediation profiles.

■ The usage data files can be copied locally by an external system and then processed based on associated Data Mediation profiles.

■ A Data Mediation profile can be generated externally as an XML file and automatically loaded and used to process a corresponding data file where both the profile definition XML file and the data file are copied locally by an external system.

Note: The integration with CA NeuMICS uses this technique.

Configure the Repository Agent

You can configure the CA Service Repository Agent by altering the settings in the configuration file %USM_HOME%\repagent\config\repagent.cfg.

The repagent.cfg file contains the following entries:

```
usm.url=http://usm_hostname:usm_port
init.pause.ms=3000
repeat.interval.ms=900000
upload.list.file=upload_list.txt
history.file=repldhis.log
#
#ftp.host=
#ftp.user=
#ftp.password=
```

The repagent.cfg file uses the following parameters:

usm.url

Specifies the Service View URL.

init.pause.ms

Specifies the pause time (for initialization) in milliseconds when the repository agent starts up.

repeat.interval.ms

Specifies the polling interval in milliseconds. The Repository Agent polls the FTP server for any new files based on this value. This value also determines the polling interval for processing local files.

upload.list.file

Specifies the profile configuration file name.

history.file

Specifies the repository agent upload history log file name. The repository agent keeps a log of all files imported into the database.

ftp.host

Specifies the FTP host name.

Note: If you specify this value and the next values (ftp.user and ftp.password), uncomment the line.

ftp.user

Specifies the FTP user name

ftp.password

Specifies the FTP password name

Profile List

When a profile is created and usage data is to be processed by the repository agent, the agent must be made aware of the new profile. The profile information is specified in the Profile List file named in the upload.list.file setting. This file is located in the %USM_HOME%\repagent\config folder.

The format of each record in the Profile List file is as follows:

profile_table_id, usage_file_name

profile_table_id

Specifies the numeric ID value from the Data Mediation profile Source Table column, without the leading zeroes.

usage_file_name

Specifies the full path and file name for the usage file corresponding to the profile referenced in the first parameter, for example, c:\Program Files\CA\Service Delivery\repagent\data\my_usage_data.txt.

Determine the Table ID of a Profile

You can determine the numeric ID value (profile_table_id) from the Data Mediation profile Source Table column.

To determine the table ID of a profile

1. Click Administration, Data Mediation, Profile Management.

    The Profile Management list of profiles appears.

2. Click the double-arrow Show More Columns icon.

    Additional columns appear.

3. The Source Table column contain entries similar to the following:

    usm_mr_itemp_001012

    The *profile_table_id* entry in the Profile List file for this profile is 1012. The usm_mr_itemp part of the table name, underscores and leading zeros are not included in the *profile_table_id* field in the Profile List file.

Note: After updating the Profile List file, restart the CA Service Repository Agent service.

Automatic Profile Load with Data Load

If the profile to be used is not already in the database, the repository agent can automatically load the profile and load the associated data. This approach requires a pair of files in the %USM_HOME%\repagent\data folder with the same file name except for the extension. The file with the extension of .xml contains the profile definition whereas the file with the extension of .txt contains the usage data. For example, suppose a profile definition named abc.xml exists. In that case, the repository agent loads that profile into the database and uses it to process an associated usage data file named abc.txt.

The profile definition XML file must follow a certain format. In the case of the integration with CA NeuMICS, this profile definition file is automatically generated by CA NeuMICS. To use this technique for other loading usage data from other systems, generate the profile XML file according to the supported format.

Profile XML File Format

To generate a profile from an external source, use the format in the sample profile XML file. This file resides in the %USM_HOME%\repagent\data\samples folder.

The profile XML file used by the repository agent to create a Data Mediation profile contains two sections:

Profile section

Contains information about the data mediation profile. Only one profile section can exist for each XML file.

- profile_name - the profile name (mandatory).

- profile_type - the profile type where 0=reference, 1=metric (default is 0).

- import_format - the format of the data source where 0=delimiter-separated file, 1=fixed length file (default is 0).

- field_separator - the delimiter between fields in the usage file. Valid values are either the character itself or the ASCII numeric value for the following characters: ampersand (&), asterisk (*), at sign (@), comma (,), dollar sign ($), exclamation mark (!), percent (%), period (.), pipe (|) or space ( ). In addition, the ASCII numeric value for a tab may be used.

Field section

Contains information about each column of the data file. You can use this section to create the fields in the Data Mediation profile. Each XML file can contain many field sections.

- field_name - database table column name (mandatory)

- display_name - display name of the field

- mandatory - mandatory status
    - 0 - neither server nor client mandatory
    - 1 - server mandatory
    - 2 - client mandatory
    - 3 - both client and server mandatory
- data_type - data type
    - 0 - string
    - 1 - integer
    - 2 - float
    - 3 - date
    - 4 - decimal
    - 5 - double
    - 6 - binary

        Note: A data type of "double" is converted to "float" during import.

- data_length - the length of this field
- data_format - the format of date data (only if data_type =3).

    A slash (/) can be replaced by a dash (-) in the formats shown below. The separator between the date and time portion can be a slash (/), dash (-) or space. Upper or lower case letters can be used. For example, YYYY-MM-DD hh24:mi:ss is a valid format.

    - MM/DD/YY
    - MM/DD/YYYY
    - MM/DD/YYYY HH:MI:SS
    - MM/DD/YYYY HH24:MI:SS
    - MM/DD/YYYY HH:MI:SS.MSS
    - MM/DD/YYYY HH24:MI:SS.MSS
    - DD/MM/YY
    - DD/MM/YYYY
    - DD/MM/YYYY HH:MI:SS
    - DD/MM/YYYY HH24:MI:SS
    - DD/MM/YYYY HH:MI:SS.MSS
    - DD/MM/YYYY HH24:MI:SS.MSS
    - YY/MM/DD
    - YYYY/MM/DD

- YYYY/MM/DD HH:MI:SS

- YYYY/MM/DD HH24:MI:SS

- YYYY/MM/DD HH:MI:SS.MSS

- YYYY/MM/DD HH24:MI:SS.MSS

- default_value - the value for this field. The Catalog system uses this value, not the input record.

- start_position - if import_format=1 (fixed length file), start_position is the starting position of the field in each record, starting at 1. If import_format=0 (delimiter-separated file), it is the field position starting with 1.

    For example:

    - For a fixed length file, a record contains "abc001" and this field is the numeric portion. Therefore, the start_position is 3, and the end_position is 6.

    - For a delimiter-separated file, a record contains "abc,001" and this field is the numeric portion. Therefore, the start_position is 2, because this field is the second field. Also, the end_position is blank.

- end_position - required only when import_format=1. It is the ending position of a field in each record.

- status - status of the field (default is 1)

    - 0 - system (cannot delete)

    - 1 - active

    - 2 - inactive

Chapter 12: Using Policies to Manage Requests

This section contains the following topics:

Overview

If your CA Service Catalog implementation uses CA Process Automation for approving and fulfilling requests, CA Service Catalog administrators can optionally create policies. Policies define conditions for automatically designating specific users as the assignees for a task during the request lifecycle. The most common use is to define conditions for automatically assigning specific approvers for services and service options in a request pending action.

This ability provides flexibility to assign tasks (such as approval or rejection of requests pending action) to users *other than* the managers of the requestor. However, you can optionally include these managers in your list of assignees.

The policy that you create consists of the following main components:

- The condition under which the policy applies

   If the condition is not met, the policy is not activated.

- The assignment to occur if the condition is met

   If the condition is met, the user or users you specify are assigned to perform a pending action, such as approval or fulfillment.

   You can specify the list of assignees according to any of several attributes, including but not limited to, the following:

   - User name (first name and last name)

   - Membership in a user group

   - Management hierarchy, meaning the manager of the user, the manager of that manager, and so forth

- An active-or-inactive setting

- Description and priority fields

Note: CA Workflow does *not* support policy driven request management.

Comparison to Other Approval Processes

CA Service Catalog offers the following options for the approval process you use to approve, reject, and fulfill requests for services or service options. When a user submits a request, approvers must approve or reject each service in the request. A catalog service uses one of the following approval processes:

No Approval

When a request containing the service is submitted, the status is changed to Approval Done.

System Approval Process

The system approval process uses a combination of the following to determine the approver and the number of approval levels:

- The management hierarchy

- The authorization level (see page 84) of each user in the hierarchy

- The approval level specified for the service

When a request containing the service is submitted and the Requested For value is a user, not an account, the Catalog system does the following:

1. Finds the Requested For authorization level of the user in the user profile.

2. Compares it with the approval level of the service.

If the authorization level of the user matches or exceeds the approval level specified by the service, then no further approval is needed. The system moves the request to the next status, typically Pending Fulfillment or Completed.

Otherwise, then the system determines an approver, as follows:

- The Requested For value is a user. In this case, the system determines the manager of the user and assigns the request to that manager for approval.

- The user does not have a manager or the Requested For value is an account. In this case, the system assigns the approval task to the user-specified in the configuration option named Default User For Request Actions.

After the manager or other approver approves the service, the system uses similar logic to determine whether another level of approval is required. If yes, then the system routes the request to a request manager whose authorization level matches or exceeds the approval level of the service.

Workflow Driven Approval Process

The workflow driven approval process uses either a CA Process Automation process or a CA Workflow process definition to determine the approval process. This option provides the most flexibility. The business logic to determine the approver and the number of approval levels is specified in a customizable process or process definition. CA Service Catalog provides sample processes and process definitions, including default ones for a single level of manager approval.

Policy driven approval

If you are using CA Process Automation to approve and fulfill requests, you can use policy driven approval. In policy driven approval, you specify conditions in policies, based on the attributes of service options, services, requested items, users, and so forth. If a policy is active and a submitted request meets the condition in the policy, then the following occurs: The users (assignees) named in the policy receive a request pending action to approve, reject, or fulfill a service option, service, or request.

Policy driven approval and system approval use a few common terms. For example, in both methods, the *level of approval* refers to the authority of an approver in numeric terms: the higher the number, the greater the authority of the approver. However, in policy driven approval, the administrator assigns each approver and authority level uniquely, with no relation to system approval.

If a policy does not apply to a request, the Catalog system uses the approval flow defined in the workflow driven approval process. For example, suppose you are using the predefined workflow approval process and no predefined sample policy applies to a request pending action. In that case, the Catalog system assigns request pending action to the manager of the Requested For user. If the user has no manager, then the system assigns the request pending action to the Default User for Request Actions This user is specified in the Service Builder Configuration.

Note: CA Workflow does not support policy driven approval.

Comparison to Events, Rules, and Actions

The conditions and actions that you specify in policies are similar to the rules and actions (see page 26) that CA Service Catalog supplies in events. You can view and edit these events, rules, and actions by selecting Administration, Tools.

The major similarly is that in both cases, you specify conditions that become part of the request lifecycle workflow in CA Service Catalog. In both cases, when a request meets the specified conditions, CA Service Catalog assigns the users specified in the policy. For example, suppose you create a policy that applies only to requests with cost greater than $100. When a user submits a request that meets this condition, the following occurs: The assignees specified in the policy are assigned a pending action to approve, reject, or fulfill the request.

Policies also have several differences with events, rules, and actions. The major difference is that events, rules, and actions are systemwide while you can optionally specify policies to be either systemwide or business unit-specific.

How to Create Policies

To create a policy, follow this process:

1. Become familiar with the Policy Builder (see page 456).

2. Create and maintain folders (see page 458) to store policies.

   Organize folders according to business units and child business units, using intuitive names.

3. Create the policy (see page 459).

   Important! Intuitive names and meaningful descriptions are critical for efficient use and reuse of policies across your organization!

   When creating the policy, you decide whether to activate it now or later, and you define its priority in comparison to other policies.

4. Create the condition (see page 461).

   When a condition you specify is met, CA Service Catalog assigns the related pending actions. The pending action is typically to approve, reject, or fulfill a requested item.

5. Enable rules and actions for policies (see page 503).

6. Test the policy by placing requests that meet its condition; that is, requests to which the policy applies.

   Click the Tracking tab for the test request to see which policies were used and what assignments were made.

7. Refine the policy, based on the test results, experience over time, or both.

Policy Builder

Administrators use the Policy Builder to create and maintain policies. To open the Policy Builder, click Service Builder, Policy Builder. A sample screen follows:

Policy List

Lists and organizes all the policies by categories in folders.

Condition Builder

Builds conditions and creates the approver list.

Attributes

Specifies details about the policies.

Use the toolbar to do the following:

- Create and maintain folders (see page 458)

- Create and maintain policies

- Test a policy

How Policies Apply to Individual Requests

CA Service Catalog matches policies to individual requests according to priority and business unit (tenant) level in the following order. A *match* means that the request or one of its contents (such as a service, service option, or form) meets a condition (see page 461) in the policy. Consequently, CA Service Catalog assigns the request pending action triggered by the condition to one or more assignees specified by the policy (see page 499).

1. High-priority policies at the lowest level business unit (farthest from the root)

2. High-priority policies at the next lowest level business unit

3. High-priority policies at the remaining business unit levels, from the lowest though the highest (the root)

4. Low-priority policies at the lowest level business unit

5. Low-priority policies at the next lowest level business unit

6. Low-priority policies at the remaining business unit levels, from the lowest though the highest

7. Once CA Service Catalog matches a policy to a request, it considers other policies *only* if they are at the same priority and business unit level.

8. If CA Service Catalog finds two or more policies at the same priority and business unit level, it does the following: It assigns requests pending action to *all* applicable assignees from *all* applicable policies.

   The Catalog system requires *all* approvals from these assignees to advance the request pending action to the next status in the request lifecycle. To help avoid situations in which multiple policies and approvers apply to a single request pending action, create and maintain conditions (see page 461) to use specific criteria.

Create and Maintain Folders

You can create and maintain folders (as needed) to store your policies. You can create, rename, move, copy, and delete folders. Organize folders to meet your requirements. As a best practice, organize policies in folders according to the business units and child -business units to which they apply. Similarly, suppose the policies apply to all business units. In that case, you can organize them according to categories under a folder named For All Business Units or something similar.

To create and maintain folders

1. Click Service Builder, Policy Builder.

   The main Policy Builder (see page 456) page appears.

   Note: If any of the remaining steps do not apply, skip it.

2. Add a folder, as follows:

   a. Select the folder to which you want to add the new folder and click Add, Folder.

   For example, to add a folder to the root folder (the Policies folder), select it and click Add Folder.

   The Add Folder dialog appears.

   b. Enter the name of the new folder and click OK.

   Note: The Add button is enabled only when it is applicable for the currently selected folder.

   The folder is created and appears under the parent folder.

3. Rename a folder, as follows:

   a. Select the folder to rename and click the Edit (pencil) icon.

   For example, to rename the "Oranges" folder, select it and click the Edit icon.

   The Rename dialog appears.

   b. Enter the new name of the folder and click OK.

   Note: Folder names are case-aware, but not case-sensitive. Thus, when you rename a folder, change more than the case. For example, you cannot rename an existing folder from spg policies to Spg Policies or SPG POLICIES.

   The folder is renamed.

4. Copy a folder, as follows:

    a. Select the folder to copy and click the Copy icon.

        For example, to copy the "Apples" folder, select it and click the Copy icon.

        The folder is copied.

    b. Select the folder that you want to be the new parent folder for the folder you are copying and click the Paste icon.

        The folder is pasted.

    The folder is copied to the new location.

5. Move a folder, as follows:

    a. Select the folder to move and click the Cut (scissors) icon.

        For example, to move the "Bananas" folder, select it and click the Cut icon.

    b. Select the folder that you want to be the new parent folder for the folder you are moving and click the Paste icon.

        The folder is pasted.

    The folder is moved to the new location you selected.

6. Delete a folder, as follows:

    a. Select the folder to delete and click the Delete (trash can) icon.

        For example, to delete the "Temp" folder, select it and click the Delete icon.

    b. When prompted, confirm the deletion.

    The folder is deleted.

You have created and maintained folders.

Create a Policy

Creating the policy is a required task in the process of creating and maintaining policies to achieve policy driven request management. A policy can apply to an individual request (see page 457).

To create a policy

1. Click Service Builder, Policy Builder.

    The main Policy Builder (see page 456) page appears.

2. Select the folder to which you want to add the new policy and click Add, Policy.

    Note: If the policy applies to a specific business unit *only*, create the policy in the most appropriate folder for that business unit.

    The Add Policy dialog appears.

3. Enter an intuitive name for the policy and click OK.

   Examples include For Laptops under $1,000, For smart phones in the Finance Department, or For All Virtual Machine Requests.

   The policy is created and appears under the parent folder. The policy fields appear, ready for your input.

4. Enter meaningful details in the Description field.

   An example follows: "Requests of laptops that cost $1,000 or more require approval of the Vice President of Financial Operations."

   Alternatively, you can specify descriptions emphasizing assignees rather than conditions. An example follows: "Requires approval of the Vice President of Financial Operations: Requests of laptops over $1,000."

   Important! Intuitive policy names and meaningful descriptions are critical for efficient use and reuse of policies across your organization!

5. Select either High or Low in the Priority drop-down list.

   Note: When a user submits a request, CA Service Catalog checks for and applies any matching *high-priority* policies. The policy applies if it contains a condition that the request meets. Examples include a laptop costing over $1,000 and a request from a user in the Facilities user group or in the Western Sales region. Only if no high-priority policy applies to the request, the system checks for and applies any matching *low-priority* policies.

6. Select either Active or Inactive in the Status drop-down list.

   ■ If you are ready to use a policy, select Active.

   ■ You can, for example, select Inactive to save policies that are either incomplete or not ready for use by a business unit.

7. Specify the condition (see page 461) for the policy.

8. Specify assignees (see page 499).

You have created the policy and are ready to complete its details.

How to Create a Condition

The condition is the major decision point of the policy. If the condition is met, the Catalog system assigns the pending action to the assignees, typically to approve, reject, or fulfill a requested item. You specify the condition using the attributes of CA Service Catalog elements such as users, requests, services, business units, and so forth. In addition, you can use <u>match functions</u> (see page 490) for creating conditions based on service options and service option elements.

Create simple conditions based on known attributes, such as category, external_id, code, item type, cost, status, and so forth. In the condition, you specify the criteria that the value of the specified attribute must meet to assign the pending action.

To create a condition, follow this process:

1. When you <u>create a policy</u> (see page 459) or edit one, you *decide* the condition for assigning the pending action to the approvers or fulfillers you specify. Examples follow:

   ■ The Asset Management group must approve all laptop requests over $1,000

   ■ The manager of the Requested For user must always approve the request

   ■ If the Requested For user belongs to a specific business unit, specify the approvers or fulfillers for that business unit in the list of assignees

   Note: As a best practice, write meaningful details in the Description field.

2. Create a condition based on your decision by entering your criteria as an expression in the Condition field.

   The condition builder is the tool in the Condition field that helps you specify valid conditions, one segment at a time. When you initially move the cursor to the field, the condition builder prompts you with valid options for the first part of the condition. These options appear in a drop-down list under the Condition field; select the option you want from the list to populate the field. As you complete each part of the condition, the condition builder continues to prompt you with valid options for the next part. This process continues until the condition is finished, typically with a closing parenthesis.

   The condition must be a valid JavaScript expression, as explained and illustrated in this topic and related topics in this documentation.

Typically specify one condition per policy, using the following format:

$(_.*group*.*attribute* *operator* '*value*')

group

> Specifies service, request, business unit, or any other group illustrated in the types of conditions linked at the end of this step.

attribute

> Specifies any attribute of that group

operator

> Specifies one of the following:

– == (equal to)

– != (not equal to)

– > (greater than)

– < (less than)

– >= (greater than or equal to)

– <= (greater than or equal to)

value

> Specifies a literal value, typically the name of a business unit, request, service, service option group, or user.

> Enter numeric values without quotation marks, for example: $(_.request.bu.status==0)

> Enclose string values in single quotation mark; for example: $(_.request.bu.taxRegion =='South')

> If a string value includes a single or double quotation mark, precede that mark with a backslash (\) as the "escape" character. For example, if the service name is Demandes d'IP Statique, then specify the condition as follows: $(_.service.name=='Demandes d\'IP statique')

As you construct an expression in the Condition Builder, the data type (string or numeric) of the attribute appears on the right, letting you know whether to use quotation marks around the value.

For example: $(_.service.name=='Procure Server')

This condition means that when the name of the service is Procure Server, the users you specify are assigned as actors, typically approvers or fulfillers.

For example: $(_.request.estimatedCost >==1000)

This condition assigns the pending action to the specified approvers or fulfillers when the estimated cost of the total request is greater than or equal to $1,000.

As a best practice, write the expression to be as simple as possible. However, if necessary, specify multiple conditions, such as compound or complex conditions, in the same expression. Use the following logical operators:

■   && for *and*

■ || for *or*

■   ! for *not*

Specify one of the following types of conditions. The linked sections explain how to specify the complete condition.

■   Conditions based on the attributes of requests (see page 463)

■   Conditions based on the attributes of users (see page 466)

■   Conditions based on the attributes of the business unit   (see page 473)

■   Conditions based on the attributes of services (see page 477)

■   Conditions based on the attributes of service option groups (see page 479)

■   Conditions based on the attributes of service options (see page 480)

■   Conditions based on the attributes of service option elements (see page 485)

■   Match functions for service options and service option elements (see page 490)

■   Conditions based on the fields of Form Designer forms (see page 491)

■   Conditions based on the attributes of the location (see page 494)

3.   Save your selections.

4.   Perform the remaining steps to finish creating the policy (see page 459).

Conditions Based on the Attributes of Requests

You can specify conditions based on the following attributes of the request affected by the policy:

completionDate	priority
dateCreated	name
dateRequired	requestedBy
description	requestedByAccountId
estimatedCost	requestedFor
id	requestedForAccountId
lastModified	status

The following attributes may require explanation:

estimatedCost

Specifies the total estimated cost of all services (including all service options) in the request. The Catalog system calculates this cost when the request is submitted.

To find the cost of an entire request, select Home, Requests and view the details for the request of interest.

priority

Specifies the priority of the request as a number, using one of the following values:

1 = high

2 = medium_high

3 = medium

4 = medium_low

5 = low

status

Specifies the numeric value of the status of the request.

To use request status as a condition for *approval*, specify this attribute in your condition, using the *approval range*, by default, less than 800. For example, the following condition is met when the service is named "Procure Laptop" and the request status is approved:

$(_.service.name=='Procure Laptop' && _.request.status < 800)

To use request status as a condition for *fulfillment*, specify this attribute in your condition, using the *fulfillment range*, by default, greater than or equal to 999. For example, the following condition is met when the service is named "Procure Laptop" and the request status is fulfilled:

$(_.service.name=='Procure Laptop' && _.request.status >= 999)

If your organization is *not* using custom statuses, you can specify the default status values (see page 572).

If your organization *is* using custom statuses, find all *actual* status values (both default and custom). To do so, open the file named requestshared.xml and record the values that you want to use in your conditions.

See also the other examples later in this topic.

Note: This file can be different based on the language of the operating system and is located in a different folder for each localized version of CA Service Catalog. For example, for English (icusen), the requestshared.xml file is located in the %USM_HOME%\view\webapps\usm\locale\icusen\request folder. For more details about this file, see the *Implementation Guide*.

other attributes

You can view most of the other attributes when you view the request list pages (see page 583). Otherwise, open a request to view its additional details.

Examples

Sample conditions follow.

■ To assign a pending action when the estimatedCost of the request equals 100 monetary units (by default, dollars), use this condition:

$(_.request.estimatedCost == 100)

■ To assign a pending action when the priority of the request equals 1, meaning high priority, use this condition:

$(_.request.priority==1)

■ To assign a pending action when a request item has not been approved (status 800) and the Requested For user has a manager, use this condition:

$(anySoWith('status', lt, 800) &&_.request.requestedForUser.manager != '')

■ To assign a pending action when a request item has been approved or when the Requested For user does not have a manager, use this condition:

$(anySoWith('status', gteq, 800) || _.request.requestedForUser.manager == '')

The assignee list for this example can specify the default user (spadmin) or another suitable user.

■ To assign a pending action when the request status is greater than 200 (submitted) and the business unit of the assignee is ca.com, use this condition:

$(_.request.status>200 && _.request.bu.id=='ca.com')

Conditions Based on the Attributes of Users

You can specify conditions based on the attributes of users, as follows:

■ Conditions based on the attributes of the Requested For user (see page 471)

The Requested For user is the user who receives the requested services.

■ Conditions based on the attributes of the Requested By user (see page 472)

The Requested By user is the user who created and submitted the request.

The Requested For user and Requested By user can be the same user or different users. If you submit a request for yourself, you are both the Requested For user and the Requested By user. If you submit a request for another user, you are the Requested By user, and the other user is the Requested For user.

■ Conditions based on the attributes of the assigner (see page 472)

Typically, the assigner is a workflow process utilizing an automated user like CERT-Service Delivery. Relatively few conditions are likely to use these attributes.

You can use the following attributes in conditions based on the attributes of users:

alias	localeCountry
commonName	localeLanguage
defaultDomain	manager
defaultRole	middleName
delegate	mobile
description	pager
email	phone
fax	roles
firstName	status
groups (user groups)	timezone
id	title
lastName	uuid

The following attributes may require explanation:

defaultDomain

Each user has a default domain (business unit) in the user profile (see page 80). Administrators set this default business unit when they add (see page 81) or edit (see page 86) the user.

Specifies different values for different parameters, as follows:

■   For the Requested For user parameter, specifies the default business unit of the Requested For user

■   For the Requested By user parameter, specifies the business unit of the request

Enter the ID of the business unit you want from the usm_tenant_ext table.

For example, to list the values of all business unit IDs in that table, run the following query on the MDB from your database client:

select tenant_id from usm_tenant_ext

To list the default business unit of a user, run the following query on the MDB from your database client:

select domain from usm_contact_domain_role where user_id=*userid* and default_domain=1

userid

Specifies the user ID whose default business unit you want to find.

default_domain=1

Specifies the default business unit of the user.

defaultRole

Specifies the default role of a user in a domain.

Enter the ID of the role ID you want from the usm_role table.

For example, to list the IDs of all roles in that table, run the following query on the MDB from your database client:

select role_id from usm_role

To list the default role of a user in a domain, run the following query on the MDB from your database client:

select role_id from usm_contact_domain_role where user_id=*userid* and default_domain=1

userid

Specifies the user ID whose default domain you want to find, for example:

select role...where user_id='john smith'...

If the user ID includes one or more spaces, enclose it with single quotation marks, as shown in the example on the previous line.

default_domain=1

Specifies that the role is for the default domain of the user.

delegate

Specifies the user ID of a delegate for auto-delegation of the requests pending action of the user. Such delegates appear in the user profile (see page 80) in the Request Auto Delegation section. Users and their administrators can specify such delegates in the profile.

If any delegate specified in the condition matches any delegate of the Requested For user, the Catalog system assigns the pending action.

Enter the user ID of the delegate you want from the MDB.

For example, to list the delegates for a user ID, run the following query on the MDB from your database client:

select delegate_id from usm_request_Auto_delegation where delegator_id=*userid* and delegation_type=0

userid

Specifies the user ID of the delegator.

delegation_type=0

Specifies that the type of delegation is auto-delegation.

Contains all the CA EEM groups to which the user belongs. You can, for example, create a policy condition to assign a request pending action based on whether a user belongs to a specific CA EEM user group.

To find the names of CA EEM user groups, query CA EEM. Log in to CA EEM and review the group names.

You can use the groups property to check if the requested by user is a member of a CA EEM group. For example, you can create a policy that requires all requests created by developers to be assigned to architects. To do so, use the following permission to check the requested by user group membership in the developers group:

_.request.requestedByUser.groups.indexOf('developers') >= 0

In contrast, you can also create a policy for users who are not members of a CA EEM group. To do so, specify the following expression:

_.request.requestedForUser.groups.indexOf('developers') < 0

localeCountry

Specifies the ISO 3166 two-letter country code of the logged in user.

For a complete list of these codes, see the website of the University of Berlin. (http://userpage.chemie.fu-berlin.de/).

At publication time, the direct link to the web page containing this list is http://userpage.chemie.fu-berlin.de/diverse/doc/ISO_3166.html.

Commonly used ISO 3166 two-letter country codes include the following:

Brazil – BR

China – CN

France – FR

Germany – DE

Italy – IT

Japan – JP

Spain – ES

United Kingdom – GB

United States – US

Note: The *localeCountry* attribute uses different values than the *country* attribute (from the ca_country table) used in several other conditions. Be careful to specify the correct attribute with the correct values in each condition you create.

manager

Enter the value of the manager user ID you want from the ca_contact table.

For example, to list the values of all manager user IDs in that table, run the following query on the MDB from your database client:

select supervisor_contact_uuid from ca_contact

status

Specifies the status of the Requested For user, as follows:

0 - Active

1 - inactive (deleted)

timezone

Specifies the code for the time zone (such as Eastern USA, Greenwich Mean Time, Amazon Time, and so forth) for the business unit.

Enter the time_zone_code from the ca_time_zone table.

For example, run the following query on the MDB from your database client:

select time_zone_code from ca_time_zone

uuid

Enter the value of the contact_uuid you want from the ca_contact table.

For example, run the following query on the MDB from your database client:

select contact_uuid from ca_contact

Title

Enter the value of the job title for the user. The ca_contact table stores the job title of each user.

For example, list the job title of a specific user (here, Omar PE Patel) in that table. To do so, run the following query on the MDB from your database client:

Select job_title from ca_contact where userid='Omar PE Patel'

To list the values of all job titles available in CA Service Catalog, run the following query:

select id from ca_job_title

Conditions Based on the Attributes of the Requested For User

To specify conditions based on the attributes of the Requested For user affected by the policy, use the following format (without spaces):

$(_.request.requestedForUser.*attribute operator* '*value*'')

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- To assign a pending action when the country of the business unit of the Requested For User is Brazil, use this condition:

- $(_.request.requestedForUser.localeCountry==BR)

  Note: BR is the ISO 3166 two-letter country code (see page 466) for Brazil.

- To assign a pending action when the role of the Requested For user is request manager, use this condition:

  $(_.request.requestedForUser.role=='request manager')

- To assign a pending action when the title of the Requested For user is Procurement Officer, use this condition:

  $(_.request.requestedForUser.title=='Procurement Officer')

Conditions Based on the Attributes of the Requested By User

To specify conditions based on the attributes of the requested by user affected by the policy, use the following format (without spaces):

$(_.request.requestedByUser.*attribute operator* '*value*'')

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- To assign a pending action when the local language of the requested by user is Arabic, use this condition:

  $(_.request.requestedByUser.localeLanguage='Arabic')

- To assign a pending action when the time zone of the requested by user is GMT-11:00 MIT, use this condition:

  $(_.request.requestedByUser.timezone=='GMT-11:00 MIT')

  The time zone code for each user is specified in the user profile. To retrieve the time zone code associated with a specific user (Isabella Lauderos), run the following query

  select time_zone_code from usm_contact_extension where user_id='Isabella Lauderos'

Conditions Based on the Attributes of the Assigner

To specify conditions based on the attributes of the assigner, using the following format (without spaces):

$(_.user.*attribute operator* '*value*')

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- To assign the pending action when the title of the assignee is Purchasing Officer, use this condition:

  $(_.user.title=='Purchasing Officer')

- To assign the pending action when the user's default role in the business unit is either Service Delivery Administrator (spadmin) or the Super Business Unit Administrator (stadmin), use this condition:

  $(_.user.defaultRole=='spadmin' || _.user.defaultRole=='stadmin')

Conditions Based on the Attributes of the Business Unit

You can specify conditions based on the attributes of the business unit, as follows:

■ Conditions based on the attributes of the business unit of the request (see page 476); that is, on the business unit of the Requested For user

■ Conditions based on the attributes of the business unit of the assignee (see page 477)

The assignee is the logged in user attempting to complete the pending action triggered by the policy condition being met. Typically, the assignee performs an approval or fulfillment task.

In both of these types of conditions, you can use the following attributes:

business unit type	parent
country	primaryContact
currency	singleAccountMode
dateFormat	stateTaxId
decimalFormat	status
description	taxRegion
email	timeFormat
federalTaxId	timezone
id	type
name	website
loginId	data1, data2, data3, data4, data5, data6
openedDate	

The following attributes may require explanation:

Business unit type

Specifies the code for the type of the business unit. Valid values are:

TE - Tenant

ST - Super Tenant

SP -Service Provider

country

Specifies the country code for the business unit.

Enter the code for the country you want from the ca_country table.

An example follows: To list the code of a specific country (here, India) in that table, run the following query on the MDB from your database client:

Select id from ca_country where country='India'

The results show that the country code for India is 114.

To list all country codes for CA Service Catalog, run the following query on the MDB from your database client:

select id from ca_country

currency

Specifies the code for the currency unit (such as dollar, Euro, pound, and so forth) for the business unit.

Enter the value of the currency_type_code you want from the ca_currency_type table.

For example, run the following query on the MDB from your database client:

select currency_type_code from ca_currency_type

dateFormat

Specifies the date format for the business unit. Valid values are:

M/d/yyyy

M-d-yyyy

d/M/yyyy

d-M-yyyy

yyyy/M/d

yyyy-M-d

dd.MM.yyyy

decimalFormat

Specifies the decimal symbol for the business unit. Valid values are:

1 implies that the decimal symbol is a comma (,).

0 implies that the decimal symbol is a dot (.).

parent

Specifies the tenant_id of the parent business unit.

Enter the parent_tenant_id of the usm_tenant_ext table.

For example, run the following query on the MDB from your database client:

select parent_tenant_id from usm_tenant_ext

Single account mode

Specifies whether the business unit contains a single account only. Valid values are:

0 - The business unit users can have multiple accounts.

1- The business unit can have only a single account.

status

Specifies the status of the business unit (not the request). Valid values are:

0 = Inactive (Deleted)

1= Active (Open)

timeformat

Specifies the time format for the business unit. Valid values are:

HH:mm:ss

HH.mm.ss

Timezone

Specifies the code for the time zone (such as Eastern USA, Greenwich Mean Time, Amazon Time, and so forth) for the business unit.

Enter the time_zone_code from the ca_time_zone table.

For example, run the following query on the MDB from your database client:

select time_zone_code from ca_time_zone

data1, data2, data3...

Specify custom data fields, if applicable, that you have created and use.

Conditions Based on the Attributes of the Business Unit of the Request

To specify conditions based on the attributes of the business unit of the request, using the following format (without spaces):

$(_.request.bu.*attribute operator* '*value*')

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- To assign the pending action based on the status of the business unit, use this condition:

  $(_.request.bu.status==0)

  Note: The value of 0 specifies that the business unit is inactive. You can, for example, use a policy to assign all requests from an inactive business unit to a specific user.

- To assign the pending action when the tax region of the business unit of the request is named South, use this condition:

  $(_.request.bu.taxRegion =='South')

- To assign the pending action when the currency of the business unit of the request is named Euro, use this condition:

  $(_.request.bu.currency =='Euro')

Conditions Based on the Attributes of the Business Unit of the Assignee

To specify conditions based on the attributes of the business unit of the assignee, using the following format (without spaces):

$(_.user.bu.*attribute operator 'value'*)

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- To assign the request pending action when the name of the business unit of the assignee is Western Regional Sales, use this condition:

  $(_.user.bu.name=='Western Regional Sales')

- You can assign the request pending action when the value of the singleAccountMode attribute of the assignee's business unit is 1 (meaning Yes). To do so, use this condition:

  $(_.user.bu.singleAccountMode==1)

- To assign the pending action when the business unit of the assignee is located in India, use this condition:

  $(_.user.bu.location.country==114)

  This condition uses 114, because 114 is the ID number for India in the ca_country table.

Conditions Based on the Attributes of Services

You can specify conditions based on the following attributes of the service affected by the policy:

bu	estimatedCost
code	id
dateAvailable	name
dateUnavailable	status
dateCreated	website
dateCancelled	version
description	

The following attributes may require explanation:

estimatedCost

Specifies the total estimated cost of a service in a request. The Catalog system calculates this cost when the request is submitted.

To find the cost of a service in a request, select Home, Requests and view the details for the request that contains that service.

status

Specifies the numeric value of the status of the service (*not* the request), as follows:

0 - Deleted

1 - Available

2 - Unavailable

3 - Created

4 - Canceled

other attributes

You can view most of the other attributes when you add or edit a service (see page 142).

Use the following format (without spaces):

$(_.service.*attrbute operator 'value'*)

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

■ To assign a pending action based on the service name, for example, if the service name is Knowledge Management Tools, use this condition:

$(_.service.name=='Knowledge Management Tools')

■ You can assign a pending action based on the estimated cost of the service being greater than or equal to 100 monetary units. (The default unit is dollars). To do so, use this condition:

$(_.service.estimatedCost<=100.0)

■ You can assign a pending action for any service option (requested item) whose status is greater than 1000 when the service ID is 9999. To do so, use this condition:

anySoWith('status', gt, 1000) || _.service.id==9999

By default, 1000 is the value for a status of Pending Fulfillment.

Conditions Based on the Attributes of Service Option Groups

You can specify conditions based on the following attributes of the <u>service option group</u> (see page 169) (formerly *rate plan*) affected by the policy. The service option group has the following attributes

bu	description
code	estimatedCost
dateAvailable	id
dateUnavailable	name
dateCreated	status
dateCancelled	

estimatedCost

Specifies the total estimated cost of a service option group in a service in a request. The Catalog system calculates this cost when the request is submitted.

To find the cost of a specific service option group in a service in a request, select Home, Requests. View the details for the request that contains that service option group. Add the costs of all service options that belong to the service option group.

If necessary, you can verify the service options that belong to a specific service option group by clicking Service Builder, Service Option Groups. Review the details for the service option group.

status

Specifies the numeric value of the status of the service option group. The status of the service option group is the same as the status of the service (*not* the request). Valid values follow.

0 - Deleted

1 - Available

2 - Unavailable

3 - Created

4 - Canceled

Use the following format (without spaces):

$(_.sog['*sogname*'].serviceoption[rownumber].*attribute operator 'value'*)

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- You can assign a pending action based on a service option group being named Procure Server and having an estimated cost greater than 1,000 monetary units. (The default unit is dollars). To do so, use this condition:

  sog['Procure Server'].estimatedCost>1000

- You can assign a pending action based on a service option group being named Create email Account and having an estimated cost greater than or equal to 200 monetary units. To do so, use this condition:

  $(_.sog['Create email Account'].estimatedCost >=200)

- To assign a pending action based on a service option group being named New Hire Onboarding and being contained in a business unit named Eastern District. To do so, use this condition:

  sog['New Hire Onboarding'].bu=='Eastern District'

Conditions Based on the Attributes of Service Options

A service option consists of a row of service option elements. You create conditions for service options within the context of the service option group to which they belong.

For service options and service option elements, you can specify conditions that use match functions (see page 490).

You can use the following attributes to specify conditions based on service options:

category	external_id
category_class	keywords
category_subclass	status
code	track_as_asset
estimatedCost	

The following attributes may require explanation:

category, category class, and category subclass

> Specifies the values for category, category class, and category subclass from the category.xml file.

> View this file and record the values that you want to use in your conditions. This file is located in a different folder for each localized version of CA Service Catalog. For example, for English (icusen), the category.xml file is located in the %USM_HOME%\view\webapps\usm\locale\icusen\billing folder.

> This file may be different, based on the language of the operating system. For details about this file, see the *Implementation Guide*.

estimatedCost

> Specifies the estimated cost of a service option in a service in a request. The Catalog system calculates this cost when the request is submitted.

> To find the cost of a specific service option in a service in a request, select Home, Requests. View the details for the request that contains that service option.

external_id, code, and keywords

> Specifies the values for the attributes named external_id, code, and keywords.

> Service builders specify these values when defining the service option elements of a service option group. Service builders typically use these attributes to add meta information about services, especially for categorizing services.

> You can find the values of these attributes (see page 485) and record them for use in this condition.

track_as_asset

> Specifies a numeric value indicating whether to track this service option as an asset in CA APM, as follows:

> 0 - No

> 1 - Yes

> **Note:** This attribute is relevant *only* when CA Service Catalog is integrated with CA APM.

> You can verify whether a service option uses this attribute (see page 485), as follows: View the value of the Track as an Asset field on the Service Option Element Options window--Options tab (see page 196).

Service Options as Row Numbers in Service Option Groups

You identify service options in conditions by their row number in the service option group. On the CA Service Catalog GUI, you can find this row number by selecting Service Builder, Service Option Groups, and opening the service option group you want.

When you do so, each service option in the group appears in a table; each row contains one service option. In the condition, specify the row number of the service option of interest. For example, suppose row 2 contains a service option named Windows server. In that case, specify $(_.sog['*sogname*']serviceoption[2] to include this service option in the condition.

By default, serviceoption[1] specifies that the requestor intentionally selects this service option. Thus, the service does *not* include this service option automatically.

Formats

Use the following format for conditions *with* a match function (see page 490):

$(anySoWith('*attribute*',operator,'*value'*))

Use the following format (without spaces) for conditions *without* a match function:

$(_.sog['*sogname*'].serviceoption[*rownumber*] *operator 'value'*)

For both formats, enclose string values in single quotation marks, and enter numeric values without quotation marks.

Examples

Consider the following examples:

- You can specify the assignee when the name of the service option group is Procure Laptop and the category of the first service option is 1. To do so, use this condition:

  $(_.sog['Procure Laptop'].serviceoption[1].category==1)

  This condition specifies that the pending action is assigned to the approver or fulfiller when the service option group is named Procure Laptop and the first service option in it belongs to Category 1. By default, Category 1 signifies hardware.

- You can specify the assignee when both of the following are true:

  - The name of the service option group is New Hire Onboarding.

  - The estimated cost of the third service option is 30 monetary units. A sample unit is Euros.

  To do so, use this condition:

  $(_.sog['New Hire Onboarding'].serviceoption[3].estimatedCost==30.0)

- You can specify the assignee when both of the following are true:

  - The name of the service option group is Handheld Devices.

  - The estimated cost of the service option in the third row of the group is 300 monetary units.

  To do so, use this condition:

  $(_.sog['Handheld Devices'].serviceoption[3].estimatedCost==300)

Examples with the Match Function

Consider the following examples:

- You can specify the assignee when the service option group includes a service option (in any row) with a category class greater than 10. To do so, use this condition:

  $(anySoWith('category_class',gt,10))

  This condition specifies that the pending action is assigned to the approver or fulfiller when any service option belongs to a category class greater than 10. By default, a category class greater than 10 signifies that the request is not related to any IT category, such as hardware or software.

- To specify the assignee when any service option in the request includes a category subclass greater than 10, use this condition:

  anySoWith('category_subclass',gt,10)

■ To specify the assignee when the external_id (a string) of any service option ends with the string "MB," use this condition:

anySoWith('external_id',endsWith,'MB')

■ You can specify the assignee when the category of any service option in the request is greater than 10 but less than 30. To do so, use this condition:

anySoWith('category',gt,10) && anySoWith('category',lt,30)

■ You can specify the assignee when *both* of the following are true:

– The value of the external_id attribute of any service option in the request begins with the case-sensitive word "Memory"

– The request contains a service option group named Procure Server and the category attribute of its first service option (row 1) has a value of 1, signifying hardware

To do so, use this condition

anySoWith('external_id',startsWith,'Memory') && _.sog['Procure Server'].serviceoption[1].category==1

■ To specify the assignee when any service option in the request is tracked as an asset in CA APM, use this condition:

$(anySoWith('track_as_asset==1'))

Note: This attribute is relevant only when CA Service Catalog is integrated with CA APM.

Find the Values of the Attributes of a Service Option

You can find the values of the attributes of a service option when required. For example, suppose you create conditions based on the attributes of service options (see page 480). In that case, you can specify the values of the external_id, code, and keyword attributes as part of the condition.

To find the values of the attributes of a service option

1. Click Service Builder, Service Option Groups.

   The Service Option Groups page appears.

2. Click the service option group that you want to open.

   The service option group opens, displaying its service options.

3. Double-click the row of the service option that you want to open.

   The Service Option Element dialog appears.

4. Click the Options tab.

   The Service Option Element Options window--Options tab (see page 196) appears.

5. Find and optionally record the value of the attributes you want, for example, the external_id, code, and keywords.

You have found and optionally recorded the values of the attributes of a service option.

Conditions Based on the Attributes of Service Option Elements

Conditions for service option elements (see page 179) must reference the service option group and the row of the service option they belong to. To identify the service option element, use its column number in the service option row.

For service options and service option elements, you can specify conditions that use match functions (see page 490).

You can specify conditions based on the following attributes of the service option elements affected by the policy:

estimatedCost item_type

item_text status

The following attributes may require explanation:

estimatedCost

Specifies the estimated cost of a service option element in a service option in a service in a request. The Catalog system includes the cost of all service option elements in the cost of the service option to which they belong. The Catalog system calculates this cost when the request is submitted.

To find the cost of a specific service option in a service in a request, select Home, Requests. View the details for the request that contains that service option.

item_type

Specifies a [valid value](#) (see page 489) for the item type, as specified in the Type field of the service option element. For example, if the type of the service option element is CA Business Service Insight contract, the value of item_type is 5. Similarly, if the type is Form Designer form, the value of item_type is 14.

item_text

Specifies the value of the Display Text field on the service option element definition page. You can specify the condition to require either exact match or an approximate match, as follows:

■ To require an exact match, specify the condition as follows:

_.sog['ab'].serviceoption[1].soe[2].item_text=='abc'

In this format, the text must match exactly, including case and spaces. For example, suppose "Premium Laptop" is the value of the Display Text field. In that case, the value of item_text must also be 'Premium Laptop': The value *cannot* be 'premium laptop' or 'Premium Laptop'; the value *cannot* be any value except an exact match.

■ To require an approximate match, specify the condition as follows:

anySoeWith('item_text',contains,'abc')

In this format, the Display Text must contain either the same text as the item_text field or a substring of it. The Display Text is not required to match exactly and is case-sensitive.

For example, if "Premium Laptop" is the value of the Display Text field, then the value of item_text can be any of the following:

– 'Premium Laptop'

– 'Laptop'

– 'Premium'

Formats

Use the following format for conditions *with* a [match function](#) (see page 490):

$(anySoeWith('*attribute*',*operator*,'*value*'))

Use the following format (without spaces) for conditions *without* a match function:

$(_.sog[*sogname*].serviceoption[*rownum*].soe[*colnum*].*attribute operator* '*value*')

For both formats, enclose string values in single quotation marks, and enter numeric values without quotation marks.

sogname

Specifies the name of the service option group.

rownum

Specifies the row number of the service option.

colnum

Specifies the column number of the service option element.

Examples

Consider the following examples:

- You can specify the assignee for the request pending action when both of the following are true:
  - The service option group is named Reserve Virtual Machine.
  - The item type is 15 (for reservation) for the first service option element in the first service option.

  To do so, use this condition:

  $(_.sog['Reserve Virtual Machine'].serviceoption[1].soe[1].item_type==15)

  Thus, this condition is met when a service option for creating or extending a reservation meets the specified criteria.

- You can specify the assignee for the request pending action when both of the following are true:
  - The service option group is named Increase Mailbox Size.
  - The estimated cost is greater than 200 monetary units for the first service option element in the first service option. The default unit is dollars.

  To do so, use this condition:

  $(_.sog[' Increase Mailbox Size'].serviceoption[1].soe[1].estimatedCost >200)

  Thus, this condition is especially useful to specify approvers or fulfillers for service options costing more than a specified amount.

- You can specify the assignee for the request pending action when both of the following are true:
  - The service option group is named Application Hosting.
  - The estimated cost is greater than or equal to 2500 monetary units for the sixth service option element in the fifth service option.

  To do so, use this condition:

  $(_.sog['Application Hosting'].serviceoption[5].soe[6].estimatedCost>=2500)

Examples with the Match Function

Consider the following examples:

- You can specify the assignee for the request pending action when the following is true: Any service option element in the request has an estimated cost greater than 30.0 monetary units (for example, pounds). To do so, use this condition:

  $(anySoeWith('estimatedCost',gt,30.0))

- You can specify the assignee for the request pending action when the following is true: The value of the item_text attribute of any service option element in the request contains the 'share' text string. To do so, use this condition:

  anySoeWith('item_text',contains,'share')

- See the examples earlier in this topic for the item_text attribute.

Valid values for item_type

Valid values for item_type and their meanings follow:

0   Text

1   Header

2   Numeric range

3   Rate

4   Application

5   CA Business Service Insight contract

6   Numeric

7   Boolean

8   Adjustment

9   Date

10   Date range

11   Day

12   Allocation

13   CA Workflow form

14   Form Designer form

15   Reservation.

Service designers specify the item type when they complete the Service Option Element Definition window--Definition tab (see page 180). They do so as part of the process of creating or editing a service option.

Match Functions for Service Options and Service Option Elements

You can use match functions when defining conditions for service options and service option elements. Use a match function to specify an attribute, value, and relationship in a condition *without* specifying any other elements required in other conditions.

Elements required in other conditions but *not* in match functions include the following:

- The name of the request, service, or service option group

- The row number of a service option or service option element

Match functions thus provide flexibility by enabling you to specify the attribute condition across multiple components of a request.

Use the following format to specify a match function for a service option:

$(anySoWith('*attribute*',*operator*,'*value*'))

Use the following format to specify a match function for a service option element:

$(anySoeWith('*attribute*',*operator*,'*value*'))

For both formats, enclose string values in single quotation marks, and enter numeric values without quotation marks.

attribute

For service options, you can use all the same attributes (see page 480) used in conditions for service options without a match function.

For service option elements, you can use all the same attributes (see page 485) used in conditions for other service option elements without a match function.

operator

For both service options and service option elements, *operator* can be any of the following:

Operator	Meaning
eq	equal to
neq	not equal to
lt	less than
gt	greater than
lteq	less than or equal to
gteq	greater than or equal to
startsWith	starts with
contains	contains
endsWith	ends with

value

> Specifies a string or numeric value for the attribute.
>
> For details, see the examples of conditions that use the match function for service options (see page 480) and service option elements (see page 485).

Conditions Based on the Fields of Form Designer Forms

You can specify conditions based on the values of fields of Form Designer forms. Use the following format to specify such conditions:

$(_.sog[*sogname*].serviceoption[*rownum*].form[*form-name*].*value of _id attribute string-operator 'value of value attribute'*)

rownum

> Specifies the row number of the service option that includes the Form Designer form.

form-name

> Specifies the name of the Form Designer forms, from the Form Designer tree. Do *not* use the value of name attribute for the form.
>
> **Note:** CA Workflow forms are not supported.

value of _id attribute string-operator 'value of value attribute'

> Specifies the attribute operator value portion of the conditions, using the same format as other conditions (see page 461). For forms, the following requirements apply:
>
> - The *attribute* specification must be the value of the _id attribute of the field you want.
>
> - The *value* specification must be based on the value of the value attribute of the same field.
>
> Typically, the *value* specification is an option based on user input or user selection.
>
> **Important!** The Form Designer stores the values for all fields in string format, even field containing numeric input specified by users or the Catalog system. Therefore, conditions based on Form Designer fields can use *only* the comparison operators suitable for strings: equal (==) and not equal (!=). Moreover, always enclose these values in quotation marks, because they are always strings.
>
> If necessary, you can optionally combine two or more expressions that use these comparison operators, using operators for specifying complex or compound conditions (see page 461).
>
> See the following examples for details about specifying conditions based on Form Designer fields.

Examples Using a Predefined Form for Product Information

This example is based on a fictitious service option group named Insider Products that contains an actual predefined form named Product Info. To see this form, select Form Designer, expand the Forms tree, and select Product Info.

Users complete this form while requesting services to obtain services from Insider Products. This form includes a Discount Program Code field with an _id attribute whose value is discount_code and a value attribute whose default value is null. However, users can change this value by entering their discount code, if applicable.

You can specify the assignee for the request pending action based on which of the following actions the user performs:

- Leaves the Discount Program Code field empty

- Enters a discount code

To meet each specification, use the following conditions, respectively:

- For an empty field:

  $(_.sog['Insider Products'].serviceoption[1].form['Product Info'].discount_code=='null')

- For a field with user input:

- $(_.sog['Insider Products'].serviceoption[1].form['Product Info'].discount_code!='null')

Examples Using a Predefined Form with Radio Buttons

This example is based on a fictitious service option group named File Shares that contains an actual predefined form named File Share Access. To see this form, select Form Designer, expand the Forms tree, and select File Share Access. This form is useful for services that users use to request access to a network file share.

The sample conditions are based on three options for access levels to the share and the values of their _id attributes, as follows:

- None: Permits no access

- Read: Allows viewing of contained files and directories, loading of files, and executing software

- Change: Provides all read permissions plus creating, deleting, and changing the directories and files

- Full Control: Provides all change permissions, including file system changes and ownership

For all these examples, both of the following apply: The service option group is named File Shares, and the third service option includes a form named File Share Access.

The field is a select box that forces the user to select an option (using a radio button) on the form. Therefore, the form field portion of the condition uses the following format:

id_*attribute of parent field*=='*value attribute of child field*'

id_*attribute of parent field*

> Specifies the value of the id_attribute of parent field, here, the field that defines the select box.
>
> In this example, on the File Share Access form, the value is access_level. The select box appears as a radio button box on the form. Users must select one of the options by clicking its adjacent radio button.

'*value attribute of child field*'

> Specifies the value of the value attribute (*not* the _id attribute) for the child attribute (the radio button option) selected by the user.
>
> In this example, on the File Share Access form, the valid values of these fields are none, read, change, and full. These values represent the options for each level of access to the file share.

View these form attributes on the Form Designer together with the following conditions to help gain a complete understanding of the relationship between them.

The following sample conditions are based on the radio button that the user selects on the File Share Access form:

■ Use this condition to specify the assignee for the pending action when the user selects No Access:

  $(_.sog['File Shares'].serviceoption[3].form['File Share Access'].access_level =='none')

■ Use this condition to specify the assignee for the pending action when the user selects Read (for read-only access rights):

  $(_.sog['File Shares'].serviceoption[3].form['File Share Access'].access_level =='read')

■ Use this condition to specify the assignee for the pending action when the user selects Change (for read and update access rights):

  $(_.sog['File Shares'].serviceoption[3].form['File Share Access'].access_level =='change')

■ Use this condition to specify the assignee for the pending action when the user selects Full Control. Full Control provides access rights for read, update, create, delete, and so forth.

  $(_.sog['File Shares'].serviceoption[3].form['File Share Access'].access_level =='full')

Conditions Based on the Attributes of the Location

You can specify conditions based on the attributes of the location, as follows:

- Conditions based on the attributes of the location of the assignee (see page 496)

  The assignee is the logged in user attempting to complete the pending action triggered by the policy condition being met. Typically, the assignee performs an approval or fulfillment task.

- Conditions based on the attributes of the location of the business unit of the assignee (see page 497)

In both of these types of conditions, you can use the following attributes:

address	name
city	state
country	phone
county	postalCode
description	uuid
fax	

The following parameters may require explanation:

City

Specifies the code for the city (such as New York City, Tokyo, Beijing, and so forth) for the business unit of the assignee. Users enter their own values, rather than run a database query. This data is user-defined, and the MDB does not store this data.

County

Is the *user-specified* value, such as a name or code, for a county. Examples include a county in a state of the United States or in another country that uses counties as a geographical unit.

The values for this attribute are user-specified and are not validated or maintained by CA Service Catalog or the MDB.

Country

Specifies the country code for the business unit.

Enter the code for the country you want from the ca_country table.

For example, to list the code of a specific country (here, India) in that table, run the following query on the MDB from your database client:

Select id from ca_country where name="India"

The results show that the country code for India is 114.

To list all country codes for CA Service Catalog, run the following query on the MDB from your database client:

select id from ca_country

State

Applies to the 50 states in the United States, to some but not all of the United States territories, such as American Samoa. Also applies to all provinces in Canada. This attribute is null for other countries and territories.

This attribute specifies the ID for the applicable state, territory, or province for the business unit of the assigner. Examples include 7404 for California and 7434 for New York.

Enter the ID you want from the ca_state_province table.

To list the IDs for all applicable states, territories, and provinces, run the following query on the MDB from your database client:

select id from ca_state_province

Verify the two-letter symbol for the state, territory, or province from the United States Postal Service (www.usps.com). You can use that symbol to find the ID. For example, the symbol for the state of New York is NY. Therefore, to list the ID for New York, run the following query on the MDB from your database client

select id from ca_state_province where symbol='NY'

This query returns a value of 7434, which you use in your condition to assign the pending action based on the user's location being the state of New York. For details, see the sample state attributes in the conditions based on the attributes of the location of the business unit of the assignee (see page 497).

To list the IDs, symbols, and names for all applicable states, territories, and provinces, run the following query:

Select id,symbol,description from ca_state_province

The results appear similar to the following:

7400 AK Alaska
7401 AL Alabama
...

Conditions Based on the Attributes of the Location of the Assignee

To specify conditions based on the attributes of the location of the assignee, use the following format (without spaces):

$(_.user.location.*attribute operator* '*value*')

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- To assign the pending action when the name of the location of the user is Home, use this condition:

  $(_.user.location.name=='Home')

- To assign the pending action when the country of the user's location is India, use this condition:

  $(_.user.location.country==114)

  Note: The code for India in the ca_country table is 114.

- To assign the pending action when the city of the location of the user is Hamburg, use this condition:

  $(_.user.location.city=='Hamburg')

Conditions Based on the Attributes of the Location of the Business Unit of the Assignee

To specify conditions based on the attributes of the location of the business unit of the assignee, use the following format (without spaces):

$(_.bu.location.*attribute operator* '*value*')

Enclose string values in single quotation marks. Enter numeric values without quotation marks.

Examples

Sample conditions follow:

- To assign the request pending action when the location of the business unit is the state of New York in the United States, use this condition:

  $(_. bu.location.state==7434)

  7434 is the code for New York in the ca_state_province table.

- To assign the request pending action when the location of the business unit is the country of Mozambique, use this condition:

  $(_. bu.location.country==169)

  169 is the code for the country of Mozambique in the ca_country table.

Assignees for Policies

Assignees are the users who act on a request, service, or service option that is assigned to them as a request pending action. The assignees approve, reject, or fulfill the requested items. Assignees are typically request managers or other administrators who act on the requests pending action. The Catalog system assigns these requests, either to the assignees personally or to an administrative group to which they belong.

For each policy, the assignees are assigned the pending action that is triggered when the policy condition is met. The approver must approve, reject, or fulfill the request, service, or service option specified by the condition. For example, suppose the condition specifies that the pending action is triggered when both of the following are true:

■ The name of the requested service is Procure Laptop.

■ The status is Submitted (by default, 200).

In this case, the assignee can be any of the following, depending on the size and scope of the organization:

■ The person or group that approves hardware requests

■ The manager of the requestor or another manager

■ The IT Finance Group

■ The Purchasing department

You can assign only one approver or several approvers. Thus, you can assign either of the following:

■ Only one *level* of approval, such as the manager only

■ Several levels of approval, such as the manager first, followed in succession by an IT approver and a financial officer

In both cases, assign approvers based on the needs of your business. These needs can include the value of the requested item and other information important to your organization.

When you are creating a policy (see page 459) or editing one, you *decide* how many approvers and how many levels of approval to specify. Examples follow:

■ For items under $100, only the manager of the Requested For user must approve the request.

■ For items over $100, both the immediate manager and the next-level manager must approve the request, in succession.

■ For requests to purchase third-party software, the following apply:

  – First, both the immediate manager and the next-level manager must approve the request, in succession.

- Next, a member of the Software Purchasing department must approve the request.

■ If the Requested For user or the Requested By user is a delegate, include the manager of that user in the list of assignees.

■ If the Requested For user belongs to a specific business unit, specify the approvers or fulfillers for that business unit in the list of assignees.

After you decide the number and level of approvals, you specify the assignees (see page 499), using the Action Builder. The Action Builder is the section of the Policy Builder under the Condition builder. The Action Builder includes a plus sign (+) icon for adding levels of approvals and fields named Level, Requirement, Assignees, and Operation.

How to Specify Assignees

To specify assignees (see page 498) while creating a policy (see page 459) or editing one, follow this process:

1. *Decide* how many approvers and how many levels of approval to specify, based on the needs of your organization.

2. Add the first assignee by clicking the plus sign (+) in the Action Builder (the box under the Condition field).

   The first level of approval appears and opens for input, including fields named Requirement, Assignees, and so forth.

3. Complete the first Requirement field (the one next to the Level 1 column), as follows:

    In the drop-down list, select ANY or ALL.

    ANY

    Specifies the following:

    ■ The first approval or fulfillment for a requested item changes the status of the item and closes any remaining pending actions for the item. (*Item* means a service or service option.)

    ■ If all users reject or cancel an item, then the item is rejected or canceled.

    ALL

    Specifies the following:

    ■ Every assignee must approve or fulfill the requested item.

    ■ If any assignees are groups, then one member of each assigned group must act on the item.

    ■ If any assignee cancels or rejects the item, then all pending actions for the item are rejected or canceled.

    Click Save.

4. Complete the first Assignees field (the one next to the first Requirement field), as follows:

    a. Click the Search (magnifying glass) icon to display the list of valid values.

        The Find Approvers dialog appears.

    b. Select a category (User, Group, or Manager) from the Assignee Type drop-down list.

        If you select User or Group, the name of the next field remains Name Filter.

        If you select Manager, the name of the next field changes to Manager Level.

    c. Enter any applicable search criteria in the next field (Name Filter or Manager Level) and click the Search icon.

        For example:

        – To specify the direct manager of the Requested For user, select Manager as the category and enter 1 in the Manager Level field.

        – To specify the direct manager of that manager, select Manager as the category and enter 2 in the Manager Level field.

    d.   Select one or more of the users, groups, or managers displayed as approvers, and click OK.

- To select a single entry, click it.

- To select multiple entries, hold down the Ctrl key while selecting the entries you want. For example, if you select five individual users and you selected ALL, then all five must approve the pending action. If four approve but one rejects, the pending action is rejected.

  Conversely, if you select five individual users and you selected ANY, then only one must approve the pending action. As soon as one of them approves, the pending action is approved and advances to the next phase or next level of approval.

  The assignees you selected are recorded. The Find Approvers dialog closes. You return to the Action Builder page.

5. Click OK under the first row of approvers, and click Save (above the Policy tree).

   You have specified and saved the first assignees for the first level of approval.

6. Specify one or more assignees, as follows:

   If you specify multiple assignees, specify whether the pending actions are to be sequential or parallel.

   Parallel actions occur at the same time and at the same level of approval.

   Sequential actions occur in order, one after the other, and at different levels of approval.

   - To specify a complex (OR) or compound (AND) expression for the first level approval, double-click the first row.

     The Operation field and the second Requirements and Assignees fields open for input.

     a.   Complete the second Assignees field as described for first Assignees field earlier in these steps.

     b.   Complete the second Requirements field as described for first Requirements field earlier in these steps.

     c.   Select AND (for a compound approval) or OR (for a complex approval) in the Operation drop-down list.

     Specify AND to require that *both* the first and second assignees you specify must approve (or fulfill) the pending action. Otherwise, the pending action is rejected (or is not fulfilled).

     Specify OR to require that *only one* of the first and second assignees you specify must approve (or fulfill) the pending action. Otherwise, the pending action is rejected (or is not fulfilled).

d. Click OK under the first row of approvers, and click Save (above the Policy tree).

If you are finished specifying approval levels and approvers, navigate away from this window.

■ To specify a second level of approval, go to the next step.

Optionally complete the steps for the previous bullet (if applicable) before going to the next step.

■ If you are finished specifying approval levels and approvers, navigate away from this window.

7. (Optional) Click the plus sign (+) in the Action Builder to add a second level of approval.

A second approval row appears, with the fields open for input.

a. Complete the first Assignees and Requirements fields in this row, as described for these fields earlier in these steps.

b. If applicable, specify a complex (either-or) or compound (both-and) expression for this row, as described in the previous step.

8. (Optional) Specify a third or more level of approval, until you are finished specifying approvers for this policy. Use the previous steps as a guideline.

9. Verify that the text in the Description field accurately represents the number and types of approvers and the levels of approval you want. Such text helps other users to decide whether to use this policy for their requests, services, or service options. Such text is also helpful for organization and maintenance.

You have assigned approvers for this policy.

The Catalog system searches for and assigns approvers and fulfillers in the following order of preference:

1. A relevant policy

2. The management hierarchy of the requestor

3. The Service Provider administrator (spadmin)

Enable Rules and Actions for Policies

Enabling the rules and actions for policies is a required task to set up policy driven request management (see page 451). You activate the policies that you have set up by enabling either the Launch Policy Driven Approval SRF (for CA Process Automation).

To enable the rules and actions for policies

1. Click Administration, Tools, Events-Rules-Actions.

    The Events-Rules-Action window appears.

2. Click the Event Type named Request/Subscription Item Change.

    Its details appear.

3. Select the rule named When Status is Submitted and Approval Process is driven by Workflow.

    The Rule Details page for this rule appears. This rule is enabled by default. If it has been disabled, click the Enable button; this button appears on the Rules bar.

4. Select the action named Launch Policy Driven Approval SRF and click the Enable button; this button appears on the Actions bar.

    The action is enabled.

    Click Done.

    You return to the Event Details page.

5. Do the following to configure CA Service Catalog so that you can use policies to fulfill requests, using *simple* fulfillment:

   a. Open the When Status is Pending Fulfillment rule; you do not exit the Event Type named Request/Subscription Item Change.

   b. Create a CA Process Automation action and select the Policy_Approval Start Request Form.

   c. Complete the parameters that appear.

   d. Specify the following parameters as indicated:

      RequestStatusUponAssignment =1000

      Specifies that the requested item is set to this status (Pending Fulfillment) when the request pending action is assigned.

      RequestStatusUponSuccess =2000

      Specifies that the requested item is set to this status (Fulfilled) when the request pending action is completed.

   The Policy_Approval Start Request Form supports *simple* fulfillment but does *not* support multilevel fulfillment and complex fulfillment. However, you can optionally create a CA Process Automation process that *does* support multilevel fulfillment and complex fulfillment. To do so, use the assignPolicyBasedPendingActions method in the web service (see page 123) named RequestService.

   The rules and actions are enabled.

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 polices 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.

Chapter 13: Using CA Service Catalog for Administrators

This section contains the following topics:

Introduction to Administration

Administrators are users whose role is Service Delivery Administrator, Super Business Unit Administrator, Request Manager or Administrator.

Note: Your user ID can be assigned various roles in different business units.

Administrators can use the catalog to make requests for themselves and others.

Subscriptions and Requests

You use the same service catalog for both subscriptions and requests. CA Service Catalog enables catalog users to request services. CA Service Accounting enables accounts in the catalog to have subscriptions.

Requests have a Requested For value. When you add services to your cart, the Requested For value is set to your user ID by default. As an administrator, you can change the Requested For value to be a user or account within your business unit scope. When you set the Requested For value to a user, you are also setting the value to the *user account* associated with the user.

Subscriptions apply to an account. You can optionally associate an account with a user. However, an account is not required to be associated with a user. As an administrator, you can manage subscriptions for an account within your business unit scope. Users who are not administrators can view subscriptions for their user account but cannot manage these subscriptions.

More information:

Introduction (see page 388)
Using CA Service Accounting (see page 387)
Using CA Service Catalog for Catalog Users (see page 561)

Request Life Cycle

Requests have a life cycle reflected in their status. The status of a request falls into one of the following several phases:

- Not Submitted

- Submitted

- Approval

- Fulfillment

- Completed

For the request and its services and service options to move through the request life cycle, its status must change.

The Not Submitted phase is controlled by the requesting user. A request is in the Not Submitted phase when it is still in the user's cart, if the user has saved his cart as an un-submitted request or if a submitted request has been rejected. The request does not leave the Not Submitted phase until the user submits the request or cart.

The Submitted phase is very brief and controlled by the system. Each service option in the request has a specified approval process. The submitted phase is used only until the approval process begins.

The Approval phase can be controlled by an approving user or the system, depending on the approval process for each service in the request. All services in the request must be approved (or not require approval) before the request enters the Fulfillment phase. If any services are rejected, the entire request is returned to the Not Submitted phase for the requesting user to manage.

The Fulfillment phase can be controlled by a fulfilling user or the system, depending on the fulfillment process for each service option in the request. The status of all service options in the request must be set to Fulfilled or Fulfillment Cancelled before the request enters the Completed phase.

The Completed phase can be controlled by an administrator or the system, depending on the type of service option elements requested. If CA Business Service Insight is integrated and if an Agreement service option element that depends on metering of a resource is included in the request, the status of the service option is changed to Pending Resource Assignment until a resource is assigned, then it is changed to Completed. For other types of service options elements, the status is set to Completed. After the status of a service option is set to Completed and if CA Service Accounting is installed, invoices include the service option.

Approval Processes

When a user submits a request, approvers must approve or reject each service in the request. A catalog service uses one of the following approval processes:

No Approval

When a request containing the service is submitted, the status is changed to Approval Done.

System Approval Process

The system approval process uses a combination of the following to determine the approver and the number of approval levels:

■ The management hierarchy

■ The authorization level (see page 84) of each user in the hierarchy

■ The approval level specified for the service

When a request containing the service is submitted and the Requested For value is a user, not an account, the Catalog system does the following:

1. Finds the Requested For authorization level of the user in the user profile.

2. Compares it with the approval level of the service.

If the authorization level of the user matches or exceeds the approval level specified by the service, then no further approval is needed. The system moves the request to the next status, typically Pending Fulfillment or Completed.

Otherwise, then the system determines an approver, as follows:

■ The Requested For value is a user. In this case, the system determines the manager of the user and assigns the request to that manager for approval.

■ The user does not have a manager or the Requested For value is an account. In this case, the system assigns the approval task to the user-specified in the configuration option named Default User For Request Actions.

After the manager or other approver approves the service, the system uses similar logic to determine whether another level of approval is required. If yes, then the system routes the request to a request manager whose authorization level matches or exceeds the approval level of the service.

Workflow Driven Approval Process

The workflow driven approval process uses either a CA Process Automation process or a CA Workflow process definition to determine the approval process. This option provides the most flexibility. The business logic to determine the approver and the number of approval levels is specified in a customizable process or process definition. CA Service Catalog provides sample processes and process definitions, including default ones for a single level of manager approval.

Policy driven approval

If you are using CA Process Automation to approve and fulfill requests, you can use policy driven approval. In policy driven approval, you specify conditions in policies, based on the attributes of service options, services, requested items, users, and so forth. If a policy is active and a submitted request meets the condition in the policy, then the following occurs: The users (assignees) named in the policy receive a request pending action to approve, reject, or fulfill a service option, service, or request.

Policy driven approval and system approval use a few common terms. For example, in both methods, the *level of approval* refers to the authority of an approver in numeric terms: the higher the number, the greater the authority of the approver. However, in policy driven approval, the administrator assigns each approver and authority level uniquely, with no relation to system approval.

If a policy does not apply to a request, the Catalog system uses the approval flow defined in the workflow driven approval process. For example, suppose you are using the predefined workflow approval process and no predefined sample policy applies to a request pending action. In that case, the Catalog system assigns request pending action to the manager of the Requested For user. If the user has no manager, then the system assigns the request pending action to the Default User for Request Actions This user is specified in the Service Builder Configuration.

Note: CA Workflow does not support policy driven approval.

Fulfillment Processes

Fulfillment is required for each service option included in the services in a request. When the status of a service option is set to Pending Fulfillment, it enters the fulfillment phase. When the status is set to Fulfilled or Fulfillment Cancelled the service option leaves the fulfillment phase. For each service Approval Process setting, the fulfillment phase of the requested service options can be managed separately.

- No Approval and System Approval Process - After approval, the status of the service options for the service is set to either Fulfilled or Pending Fulfillment.

- Workflow Driven Approval Process - After approval, the status of the service options for the service is set to Pending Fulfillment.

When you choose options that cause the status of the requested service options to be set to Pending Fulfillment, the fulfillment process must be managed by a Workflow process definition. Several process definitions manage the fulfillment process. These process definitions and associated event rules support three different approaches to fulfillment.

These fulfillment options may fit your business processes exactly but it is more likely that you alter the distributed components to more accurately fit your business processes. Before altering the existing components or adding to them, it is important to understand what they do.

Note: Enabling all the rules related to all three options is not recommended, becasue the three fulfillment options overlap.

Complex Fulfillment

The Complex Fulfillment process takes into account the current status value of the service option, and new status being assigned to determine the next step in the fulfillment process. It supports major fulfillment phases related to:

- Checking availability of the requested service option when initially requested or when the service option status is Received.

- Optionally opening a CA Service Desk Manager Change Order (for Hardware or Software) or notifying appropriate fulfillment personnel when the service option is found in available inventory.

- Notifying Procurement of the requested service option when it is not found in available inventory so it can be ordered and marked as Received when it arrives.

Simple Fulfillment

The Simple Fulfillment process notifies the appropriate fulfillment personnel that the item needs to be fulfilled with no special regard for fulfillment phases.

CA Service Desk Manager Request Fulfillment

The CA Service Desk Manager Request Fulfillment process opens a CA Service Desk Manager request, notifies the appropriate fulfillment personnel that the service option needs to be fulfilled and leaves the fulfillment process up to the CA Service Desk Manager request logic.

Configure the Display of Columns on Request List Pages

Administrators with the Service Provider role (by default, spadmin) can optionally configure the display of columns on request list pages (see page 583). As an administrator, you can sort the columns, configure their width, include or exclude a column from display, and set the default number of requests listed per page. CA Service Catalog applies the settings you specify to all users in all business units.

To configure the display of columns on request list pages

1. Verify that you are logged in to CA Service Catalog as a Service Provider administrator (spadmin).

2. Click Home, Requests.

    The Home, Requests page appears.

3. Click any of the following options:

    ■   Open Requests

    ■   My Recent Requests

    ■   Completed Requests

    ■   Pending My Action

    ■   Advanced Search–after you search for requests (see page 587) and view the results

    The request list page matching your selection appears.

    Note: The settings you specify in this procedure apply only to the Request List page that you select (for example, the Open Requests page or the My Recent Requests page), not to *all* list pages as a group. You cannot configure *all* Request List pages at once. However, you can configure each page *individually*, using this procedure.

4. Click the Configure button under the column heading.

   The Catalog system activates your ability to configure the column display.

5. Move the cursor over the column of interest, such as the Name, ID, or Actions column. Click the down arrow next to the column name.

   The Sort options and the Columns drop-down menu appear under the column you selected.

6. Do any or all of the following:

   ■ To include any column on the request list pages, click the name of the column in the Columns drop-down menu and verify that the column name is selected.

      For example, if the Priority column does not appear on the request list pages, you can optionally add it by clicking its name in the Columns drop-down menu and verifying that the name is selected.

   ■ To remove any column from the request list pages, click the name of the column in the Columns drop-down menu and verify that its name is not selected.

      For example, if the Date Modified column does appear on the request list pages, you can optionally remove it by clicking its name in the Columns drop-down menu and verifying that its name is not selected.

   ■ To sort the list of requests according to the ascending or descending order of any field, do the following:

      a. Move the cursor to the field of interest.

      b. Click the down arrow next to the column name

      c. Click either Sort Ascending or Sort Descending.

7. Click the arrow next to the Results-per-page drop-down list and specify the maximum number of requests that appear on each Request List page.

   The value that you specify becomes the default value for all users.

8. Click Save or Cancel.

   Your changes are either saved or cancel to discard them.

You have configured the display of columns on the Request List page you selected.

Note: Users can optionally sort columns and set the maximum number of requests per page, on individual Request List pages only. However, users cannot add or remove columns. When a user moves to a new Request List page, the default setting replaces their custom settings. Users can optionally customize the display for each new Request List page they view.

Discrete Request Life Cycle

With a discrete life cycle, you can enable services and service options to advance to higher request statuses without waiting for all other services and service options to reach the same statuses. Consequently, services and service options can reach Fulfilled and Completed statuses, and can be invoiced earlier than without a discrete life cycle, even if a request manager rejects one or more services or service options in the same request.

As an administrator, you can optionally specify discrete settings for the request life cycle for individual business units, as follows:

- Depending on the configuration you choose, you can specify the status at which request managers (approvers and fulfillers) can discretely (individually) approve, reject, or fulfill each service option in every service in the request. The setting you specify applies to the request *from* the starting status that you specify *through* the remainder of the request life cycle.

- You can optionally specify the status at which individual service options in a service and individual services in a request advance to further statuses in the request life cycle independently. Here, *independently* means *without* requiring or waiting for any other service options in the same service or any other services in the same request to advance to any further statuses. The setting you specify applies to the request *from* the starting status that you specify *through* the remainder of the request life cycle.

- You can optionally specify the effect of the rejection of a single service or service option upon other services and service options in the same request:

  - If this setting is Yes, when you reject a service or service option, the remaining services or service options can advance if they are approved.

  - If this setting is No, when you reject a service or service option, the entire request is rejected, even other services or service options that were previously approved change to Rejected status and can no longer advance in the request life cycle.

To specify these discrete settings, <u>set their configuration parameters</u> (see page 527).

Note: For all these parameters, the default settings maintain backward compatibility, so the request life cycle functions that same as it did in past releases before this parameter was implemented.

How to Implement Discrete Request Life Cycle

To implement discrete request life cycle, follow this process:

1. Consider the process automation tool you are using for approval and fulfillment—CA Process Automation, CA Workflow, or both—and proceed as follows:

   ■ The predefined CA Process Automation processes supplied for use with CA Service Catalog require *no* additional configuration to support the discrete request life cycle. However, verify that they work efficiently with the discrete request life cycle after you implement it.

   ■ The predefined CA Workflow process definitions supplied for use with CA Service Catalog *do* require additional configuration to support the discrete request life cycle.

     If you decide to use CA Workflow, use the guidelines to configure the predefined CA Workflow process definitions (see page 518) as required.

2. Decide and follow the best option for introducing the use of discrete life cycle, to minimize impact on requests in progress. Options include the following:

   ■ If you are upgrading from a previous release of CA Service Catalog, begin using a discrete life cycle in a test system, verify that the request life cycle works as expected, and move to a production system .

   ■ Instruct users to stop creating requests in your organization temporarily. As soon as most or all of the requests in progress are completed, begin using a discrete life cycle by setting the related configuration parameters, as explained later in this topic.

   ■ In-progress requests may become stuck as a result of the configuration changes. If requests become stuck, retry, or override (push-through) the alerts (see page 627) to complete the requests.

     Note: Retries and overrides may be especially required when you use the configuration setting named Allow Discrete Handling of Service Options After.

   ■ Give special attention to cancelled items, rejected items, and items pending fulfillment while others are still in pending approval, when one or both of the following are true:

     – You expect some requests to reach pending fulfillment after all the items are approved

     – You expect to reject the entire request if one of the item is rejected

     In these cases, take special care before changing the configuration to support the discrete request life cycle. Optionally, verify that such requests have completed before you make the configuration changes.

3. Review these important considerations:

- Overall status of the request

  The service option with the lowest status value (see page 572) determines the overall status of the request. To see the status of a request, select Home, Requests, open the request, and view the status in the General Information box.

  When the request contains services and service options having different status values, the lowest status value appears with asterisk (*) next to it, indicating that at least one service or service option in the same request has a higher status..

  Note: When determining the overall status of the request, the Catalog system ignores services and service options that are rejected, completed, deleted, cancelled, pending cancellation, or on hold.

- Billing

  If you are using CA Service Accounting, the invoices you generate (see page 406) *do* include completed services and service options, even if other services and service options in the same request are not completed. The uncompleted services and service options are *not* included in the same billing run as the completed ones.

- Request SLAs

  This point is relevant only if you use request SLAs (see page 253). The monitoring and recording of request SLA data is not affected when you use a discrete request life cycle. The Catalog system monitors and records the status changes of service options that have request SLAs the same way, regardless of your discrete request life cycle configuration settings.

- CA Business Service Insight-related metrics

  This point is relevant only if you are integrating CA Service Catalog with CA Business Service Insight. Your discrete request life cycle configuration settings do not affect the monitoring and recording of any CA Business Service Insight contract-related metrics associated with service options in the Catalog system.

- Sample settings and their meanings (see page 520)

- Common settings (see page 525)

- Table of discrete settings and meanings (see page 526)

4. Set the configuration parameters (see page 527) to achieve the results you want, based on the information gained from your review.

Configure CA Workflow Process Definitions

The predefined CA Workflow process definitions supplied with CA Service Catalog require additional configuration to support the discrete request life cycle.

Important! This topic provides guidelines only. While this procedure is expected to work for most users, it is not certified or supported.

To configure the predefined CA Workflow process definitions to support a discrete request life cycle

1. Note the following updates for the Web Services SOAP calls used in some CA Workflow process definitions:

Before	After
addWorkflowPendingAction	assignPendingAction
addWorkflowPendingActionWithDesc	assignPendingAction
addProcessInstanceToRequestItem	addProcessInstance
addProcessInstanceToRequestItemOffering	addProcessInstance
updateRequestItemRowStatus	updateRequestItemStatus
updateRequestOfferingStatus	updateRequestItemStatus

2. Review and configure the assignPendingAction web service method (see page 519).

3. Review the Web Services API documentation (see page 124) for the new SOAP calls listed in the After column and update the SOAP calls in your applicable CA Workflow process definitions accordingly.

4. Decide whether to use (or continue to use, if applicable) the Request Email actions to send emails.

   Typically, you use these actions in rules in either the Request Pending Action Change event (introduced in CA Service Catalog 12.6) or the Request/Subscription Item Change event (if you continue to use it from previous releases). You may also use these actions in other similar rules, especially, if applicable, any rules that you have created by copying and modifying the predefined (built-in) rules in these events.

5. If necessary, create a rule action (see page 32) for handling request emails according to your custom requirements.

6. Review your CA Workflow process definitions and delete any nodes in them that perform the same functions as your predefined or custom rule actions for handling request emails.

   For assistance, see the CA Workflow IDE online help.

Review and Configure the assignPendingAction Web Service Method

You review and configure the assignPendingAction web service method as part of the process of configuring the predefined CA Workflow process definitions supplied with CA Service Catalog to support the discrete request life cycle.

Important! This topic provides guidelines only. While this procedure is expected to work for most users, it is not certified or supported.

To review and configure the assignPendingAction web service method

1. Review thoroughly the web service API for this method, which does the following:

    ■ Adds pending actions to users and groups that are sent in the parameter named   pendingActionUsersOrGroups

    ■ If the parameter value is empty, assigns requests pending action to the manager of the requested for user

    ■ If no manager exists for requested for user, assigns requests pending action to the default user for request action

    ■ Updates the status of the request, service, or service option–whichever applies for individual requests pending action

2. Consider deleting the node in the USM_Approval process definition (or your custom version of it) that updates the Update Offering Status to Pending Approval Status. The assignPendingAction web service method is a more efficient method to performs this update.

    The decision depends on the specific logic in your USM_Approval process definition.

3. Consider deleting the nodes in the USM_Approval process definition that Add Pending for Approvers.

    If it contains multiple nodes for assigning multiple approvers, you can remove all of these nodes, because you can assign requests pending actions to multiple users and groups with the assignPendingAction web service method.

4. Consider deleting nodes that get manager info and the nodes that Use Default Request Action for User in the USM_Manager_Approval process definition. Consider doing so because the assignPendingAction web service method automatically assigns requests pending actions to the appropriate users when the pendingActionUsersOrGroups parameter of the method is empty.

5. If you choose to remove the nodes from the USM_Approval process definition in the previous step, do the following; otherwise, skip this step:

■ Verify the logic that you are using for setting the approver list in the USM_Approval process definition or the USM_Manager_Approval process definition.

■ Note that the Approval Needed for Request node in the USM_Approval process definition, which assigns the Open Document operation to Approval User, fails if the approval list mentioned in the previous bullet is empty.

For this reason, consider assigning a hard-coded Service Delivery administrator user ID, for example, spadmin, to the "Approval User" role and update the completedBy variable in the Post tab in this node with the spadministrator user ID. If you make this change, CA Workflow tasks are created for the hard-coded Service Delivery administrator user ID *only*.

You have reviewed and configured the assignPendingAction web service method.

Sample Settings and Their Meanings

Review the sample settings and their meanings for the discrete request configuration parameters named Allow Discrete Handling of Service Options After and Allow Discrete Request Life Cycle After.

For all examples, if the request manager rejects one or more services or services options in a request that includes other services and service options, the setting of the Allow Discreet Handling for Reject configuration parameter (see page 527) determines effect of the rejection on the other service options and services in the request, regardless of whether the other services and service options were previously approved.

Examples for Allow Discrete Handling of Service Options After

The Allow Discrete Handling of Service Options After configuration parameter (see page 527) specifies the status at which request managers (approvers and fulfillers) can handle requests pending action (approval, rejection, or fulfillment) discretely. When the request reaches the status you specify, request managers can discretely (individually) approve, reject, or fulfill each service option in every service in the request. Examples follow.

Example 1--Submitted

In this example, you want *both* discrete approval and discrete fulfillment. Thus, you want to approve or reject service options individually, rather than approve or reject *all* service options in the service in one action. Similarly, you want to fulfill service options individually, rather than fulfill *all* service options in the service in one action.

To achieve this goal, specify Submitted as the value for this parameter. Consequently, as soon as the user submits the request, approvers can approve or reject each service option in a service individually. Moreover, fulfillers can fulfill each approved service option individually.

Request managers may receive several notification emails when they address pending actions, if you do both of the following:

- Set the Allow Discrete Handling of Service Options After parameter to Submitted

- Keep the setting of Yes (the predefined setting) for the parameter named Notify users when they complete their own pending actions

The latter is a Service Builder, Request Management configuration option. If the latter option is set to Yes, request managers receive a notification email every time they approve, reject, or fulfill a service option, service, or request. For example, if a service has five service options and the request manager approves each option, he or she receives five confirmation emails. To prevent request managers from receiving such emails, set Notify users when they complete their own pending actions to No. For details about setting configuration options, see the *Implementation Guide*.

Example 2-- **Pending Fulfillment**

In this example, you do *not* want discrete approval but *do* want discrete fulfillment. Thus, you do *not* want to approve or reject service options individually but rather want to approve or reject all service options in the service in one action. However, once the entire request is approved, you want to *fulfill* service options individually.

To achieve this goal, specify Pending Fulfillment as the value for this parameter. Consequently, as soon as the entire request is approved, fulfillers can fulfill each approved service option in every service individually.

For backward compatibility, specify Pending Fulfillment, so that the part of the request life cycle *before* the Fulfillment status functions that same as it did in past releases of the product before this parameter was implemented.

Example 3--Completed

In this example, you do *not* want discrete approval or discrete fulfillment. Thus, you do *not* want to approve, reject, or fulfill service options individually at any point during the request life cycle. Instead, you always want to approve, reject, or fulfill all services in the request as a single unit. To achieve this goal, specify Completed as the value for this parameter.

Essentially, this setting "turns off" discrete handling of approval and fulfillment activities. Consequently, request managers must approve, reject, and fulfill all services in the request at once.

Note: The hold and resume actions continue to function for individual services and service options, because of backward compatibility, so the request life cycle functions that same as it did in past releases of the product before this parameter was implemented.

Examples for Allow Discrete Request Life Cycle After

The Allow Request Life Cycle to Continue After configuration parameter (see page 527) specifies the status at which individual service options in a service and individual services in a request advance to further statuses in the request life cycle independently. Here, *independently* means *without* requiring or waiting for any other service options in the same service or any other services in the same request to advance to any further statuses. Examples follow.

Example 1--Submitted

For example, suppose that you want each service option in a service to be able to advance to the remaining statuses in the request life cycle independently after being approved, rather than requiring or waiting for any other service options in the same service to be approved. To achieve this goal, you specify Submitted as the value for this parameter.

Consequently, as soon as the user submits a request, each service option in all services can advance independently to the remaining statuses in the request life cycle as soon as it (the service option) is approved, *without* waiting for any others to be approved or rejected.

Similarly, as soon as the user submits a request, each service in all requests can advance independently to the remaining statuses in the request life cycle as soon as it (each service) is approved *without* waiting for any others to be approved.

If an approver does not act on one or more service options, the approved service options in the same service can advance to the remaining statuses in the request life cycle. If an approver does not act on one or more services, the approved services in the same request can advance to the remaining statuses in the request life cycle.

Thus, with Submitted as the value for this parameter, the following results occur in a service that has two options: one approved and one no action, as follows:

Service Option	Approver's Action	Resulting Status
Option-1	Approve	Pending Fulfillment
Option-2	No action	Pending Approval

After all service options in the same service are approved, the service reaches Pending Fulfillment status.

After all services in the same request are approved, the request reaches Pending Fulfillment status.

Example 2--Pending Fulfillment

For example, suppose you want each service option in a service to be able to advance to the remaining statuses in the request life cycle independently after all services in the request are approved. To achieve this goal, you specify Pending Fulfillment as the value for this parameter.

Consequently, as soon as the service is approved, it reaches Pending Fulfillment, and each service option in the service can advance independently to the remaining statuses in the request life cycle independently. Similarly, as soon as the request is approved, it reaches Pending Fulfillment, and each service in the request can advance independently to the remaining statuses in the request life cycle independently.

A setting of Pending Fulfillment as the value for this parameter differs from a setting of Submitted in these ways:

- Approved service options *cannot* advance to the remaining statuses in the request life cycle *until* all other service options in the same service are approved.

- Approved services *cannot* advance to the remaining statuses in the request life cycle *until* all other services in the same request are approved.

Thus, with Pending Fulfillment as the value for this parameter, the following results occur in a service that has two options, approved sequentially, as follows:

Service Option	Approver's Action	Resulting Status
Option-1	Approve	Approval Done
Option-2	Approve	Pending Fulfillment
Option-1	Fulfilled	Completed
Option-2	Fulfilled	Completed

After all service options in the same service are approved, the service reaches Pending Fulfillment status.

After all services in the same request are approved, the request reaches Pending Fulfillment status.

Example 3--Completed

In this example, you do *not* want individual service options in a service or individual services in a request to be able to change statuses independently of each other at any point in the request life cycle.

Instead, you always want complete services and complete requests to change statuses *only* as a single unit, not independently. To achieve this goal, specify Completed the value for this parameter.

Essentially, this setting "turns off" any possibility of service options in a service or services in a request changing statuses independently of each other at any point in the request life cycle.

Note: The hold and resume actions continue to function for individual services and service options, because of backward compatibility, so the request life cycle functions that same as it did in past releases of the product before this parameter was implemented.

Common Settings

These following points apply to all combinations of the Allow Discrete Handling of Service Options After and Allow Discrete Request Life Cycle After:

- If the Allow Discrete Handling for Reject configuration parameter (see page 527) setting is No and you reject any service or service option, the entire request is rejected, and the request life cycle stops immediately. This option thus provides backward compatibility, so that the request life cycle functions that same as it did in past releases of the product before this parameter was implemented. After the rejection, the following status changes occur:
  - The status for the entire request changes to Not Submitted – Rejected.
  - Previously approved service options change to Not Submitted – Approved status.
  - All other service options move to Not Submitted – Rejected status.

  The requestor can optionally modify and resubmit the request.

  However, if this setting is Yes and you reject any service or service option, the remaining services or service options continue their request life cycle according to settings for the Allow Discrete Handling of Service Options After and Allow Discrete Request Life Cycle After parameters, as specified in these tables.

- If the request manager rejects one or more services or services options in a request that includes other services and service options, the setting of the Allow Discreet Handling for Reject configuration parameter (see page 527) determines effect of the rejection on the other service options and services in the request, regardless of whether the other services and service options were previously approved.

■ The Perform Action icon or button appears on the GUI in all cases, except when Allow Discrete Handling of Service Options After is set to Pending Fulfillment and Allow Discrete Request Life Cycle After is set to Completed. When you use these settings, the Perform Action icon or button never appears on the GUI.

When you click the Perform Action, a new page opens, enabling you to perform all applicable actions for each service or service option in the request, such as approve, reject, fulfill, transfer, delegate, take, return, and so forth.

Table of Discrete Settings and Meanings

The following table shows whether the request manager acts on a service, service option, or either one, based on the settings for Allow Discrete Handling of Service Options After (ADHSOA).

The table applies to all settings of Allow Discrete Request Life Cycle After.

Request Manager Action	ADHSOA= Submitted	ADHSOA= Pending Fulfillment	ADHSOA= Completed
Approve or Reject	service option	service	service
Hold or Resume	either	either	either
Transfer or Delegate	service option	**A**-service **B**-service option	service
Override	service option	**A**-service **B**-service option	service
Take or Return	service option	**A**-service **B**-service option	service
Push-Through	service option	**A**-service **B**-service option	service
Cancel	service	service	service
Fulfillment Cancelled	service option	service option	service
Fulfill	service option	service option	service
CA Service Desk Manager Change Order	service option	service option	service

A-service, from Submitted status through Approval Done status.
B-service option, from Pending Fulfillment status through Fulfilled status.

Note: For details about CA Service Desk Manager change orders, see the *Integration Guide*.

Set the Configuration Parameters

To enable the discrete handling of requests pending action (see page 515), set the related configuration parameters. You can use these configuration parameters individually; however, they are most effective when you use them together to achieve the request processing that you want.

To set the configuration parameters

1. Click Service Builder, Request Management Configuration.

2. Set the following parameters:

   Allow Discrete Handling of Service Options After

   Specifies the status at which request managers (approvers and fulfillers) can handle requests pending action (approval, rejection, or fulfillment) *discretely*. When the request reaches the status you specify, request managers can discretely (individually) approve, reject, or fulfill each service option in every service in the request.

   In other words, you can use this parameter to specify that pending actions are assigned at the service option level, rather than the service level.

   The setting you specify applies to the request *from* the starting status that you specify *through* the remainder of the request life cycle.

   Valid values are Submitted, Pending Fulfillment, or Completed.

   The default is Pending Fulfillment, for backward compatibility, so that the request life cycle functions the same as it did in past releases before this parameter was implemented.

   See the sample settings and their meanings (see page 520).

Allow Discrete Request Life Cycle After

Specifies the status at which individual service options in a service and individual services in a request advance to further statuses in the request life cycle independently. Here, *independently* means *without* requiring or waiting for any other service options in the same service or any other services in the same request to advance to any further statuses.

As soon as the request reaches the status that you specify, the service options that you approve or fulfill can complete the remainder of the request life cycle, even if other service options in the same service are not acted on.

Similarly, as soon as the request reaches the status that you specify, the services that you approve or fulfill can complete the remainder request life cycle, even if other services in the same request are rejected or not acted on.

Valid values are Submitted, Pending Fulfillment, or Completed.

The default is Completed, for backward compatibility, so that the request life cycle functions the same as it did in past releases of the product before this parameter was implemented.

The setting you specify applies to the request *from* the starting status that you specify *through* the remainder of the request life cycle.

See the .

Allow Discrete Handling for Reject

Specifies the effect of the rejection of a single service or service option upon other services and service options in the same request:

■ If this setting is Yes, when you reject a service or service option, the remaining services or service options can advance if they are approved. That is, other services in the same request can advance in the request life cycle if they are approved. Similarly, other service options in the same service can advance in the request life cycle if they are approved.

■ If this setting is No, when you reject a service or service option, the entire request is rejected. That is, all services in the same request are rejected and can no longer advance in the request life cycle, even if they were previously approved. Similarly, all service options in the same service are rejected and can no longer advance in the request life cycle, even if they were previously approved.

The default is No, for backward compatibility, so that the request life cycle functions the same as it did in past releases of the product before this parameter was implemented.

Request Service Level Reports

Several report data views and underlying data objects enable you to monitor request service levels.

Request Fulfillment Data View

This report asks for a date range and business unit. It displays a table where each row is for one request and there are columns to show how long it took in days, hours and minutes to approve, fulfill and complete the request. The date range is used to limit the list to just those requests whose initial creation is within the range specified. The business unit value is used to limit the report to just those requests for a user or account from a business unit.

Clicking Request ID field displays a Request Item Fulfillment report for the service options contained in the request. This sub-report displays any times as red that are longer than the violation duration for the service option and any times as yellow that are longer than the warning duration for the service option.

Request Item Fulfillment Averages Data View

This report asks for a date range and business unit. It displays a table where each row is for one service option and there are columns to show the average time in days, hours and minutes to approve, fulfill and complete the requests for this service option. The date range is used to limit the list to just those requests whose initial creation date is within the range. The business unit value is used to limit the report to just those requests for a user or account from a business unit. This report displays any times as red where the average is longer than the violation duration for the service option and any times as yellow where the average is longer than the warning duration for the service option.

Clicking Service Option field displays a Request Item Fulfillment report displaying all the requests that contain the service option. This sub-report displays any times as red that are longer than the violation duration for the service option and any times as yellow that are longer than the warning duration for the service option.

Monitor Request Service Levels

It is important to monitor how quickly requests for service options are being fulfilled. By default, there are three important phases of the request life cycle that can be monitored:

- Approval - How long it takes for the status of a requested service option to go from Submitted to Approval Done

- Fulfillment - How long it takes for the status of a requested service option to go from Approval Done to Completed

- Completion - How long it takes for the status of a requested service option to go from Submitted to Completed

For each service option, service providers can establish SLA warning and violation thresholds for each phase.

Example

This examples uses a service option named Standard Desktop. For illustration, the fulfillment phase warning time is 5 days, and the violation time is 7 days. In this way, the service provider sets the expectation that a standard desktop is to be fulfilled (configured, delivered and set up) within 5 days. Moreover, if the time to fulfillment exceeds 7 days, then a violation has occured.

The violation setting for a service option appears in the catalog as the *estimated time to fulfill* when a catalog user is browsing services. In this example, the estimated time to fulfill the Standard Desktop service is 7 days.

Note: You can set up reports to monitor other request life cycles phases by altering the STATUS_RANGES value used in a report data object.

More information:

Maintain Files when using Multiple Servers

Several features of CA Service Catalog use files on the Service View file system. If you have multiple Service View servers, you need to keep those files synchronized on all servers; otherwise, an error may occur.

The following table identifies the features that use files on the Service View server and the folders where those files are located:

Feature	Folder
Attachments	%USM_HOME%\view\documents
CA Service Catalog Images	%USM_HOME%\FileStore\images\offerings
Documents	%USM_HOME%\view\documents
CA Service Catalog Forms	%USM_HOME%\view\forms
Offline Data Objects and Data Views	%USM_HOME%\reporting\offline

PDA Approval

You can optionally use a PDA to review, approve, and reject requests.

Introduction and Benefits

When traveling, administrators, managers, and other key personnel in most organizations use a mobile device named a *personal digital assistant* (*PDA*), to communicate, using telephone calls and emails and also to connect to the internet, using a special small-screen web browser. In this way, these employees continue to keep business processes moving, despite being out of the office.

As a CA Service Catalog administrator, you can optionally configure your CA Service Catalog implementation and its services to enables users to approve and rejects requests using a PDA. Such users receive an automatic email notification on their PDAs that a request is pending approval (a request approval email).

Depending on how you set up PDA approval, PDA users can optionally use the links you supply in the request approval email to do one or more of the following actions:

■ Log in to CA Service Catalog, view the request details, and approve or reject the request

■ Approve or reject the request, service, or service option without viewing its details

   Users must enter their login credentials manually

■ Approve or reject the request, service, or service option *without* logging in to CA Service Catalog; this action is direct email approval (see page 537)

By using PDAs in this manner, these employees keep the request life cycle component of your business moving, despite being out of the office.

Approving and rejecting requests using a PDA (*PDA approval*) is similar to the same process using a laptop or desktop computer. However, PDA approval includes some additional setup work (see page 533) and includes some limitations (see page 535) to consider, when compared to the traditional process using a laptop or desktop in the office.

How to Set Up PDA Approval

Enabling PDA approval gives administrators, managers, and other key personnel the flexibility to approve and reject requests from outside the traditional office setting, for example, while traveling or working at a client site. Approving and rejecting requests using a PDA (PDA approval) is similar to the same process using a laptop or desktop (office computers). However, PDA approval includes some additional requirements and some limitations when compared to the traditional process. Complete these tasks to enable PDA approval.

1. Verify that you have installed and configured CA Service Catalog; especially verify that you have set up CA Service Catalog to approve requests using either CA Process Automation or CA Workflow.

   Note: For details about setting up CA Service Catalog to use either CA Process Automation or CA Workflow to approve requests, see the *Integration Guide*.

2. Verify that CA Service Catalog supports the mobile devices and mobile browsers that you want to use for PDA approval.

   You can approve and reject requests using the Blackberry browser on a Blackberry or the Safari browser on an iPhone. The minimum requirements for PDA devices are as follows:

   - Its email software must be able to render HTML.

   - Its browser must support basic JavaScript features and functions.

3. Configure your PDA to accept emails from CA Service Catalog. This task applies to all users, regardless of the method of PDA approval (direct email approval or logged in approval) that you chose in later steps.

   For assistance, see your PDA documentation.

4. Set the Request Management configuration parameters (see page 536) to enable PDA approval.

   When PDA approval is enabled, approvers receive request approval emails automatically on both their office computers and PDAs, to notify them that a request is pending their approval.

5. Review the limitations of PDA approval (see page 535).

6. Review direct email approval (see page 537), including its prerequisites (see page 540) and the comparison to logged in approval (see page 538). Decide which method to use for approving and rejecting requests using a PDA. As a best practice, use either method, but not both methods.

7. If you decide to use direct email approval, set up direct email approval (see page 537) to meet your requirements.

   If you decide to use logged in approval, skip the topics for setting up direct email approval.

8. Review and customize the request approval email in use at your implementation.

    If you have already installed (or upgraded) and configured CA Service Catalog, you already have an existing request approval email that approvers access through office computers. CA Service Catalog automatically uses that email for both office computers and PDAs. When you enable PDA approval through either method (direct email approval or logged in approval), you review and customize this email to achieve these goals:

    ■   Configure the text and links of the email to enable approval and rejection by either logged in approval (the default) or direct email approval

    ■   As a best practice, configure the text and links of the email to enable approval and rejection at *only one* of the following levels:

        –   The entire request only (the default)

        –   Individual services – This option applies only when you use the Notify Approver action to send the request approval email.

        –   Individual service options – This option applies only when *both* of the following are true: you use the Notify Approver action to send the request approval email and you use a discrete request life cycle with approval set at the service option level (Allow Discrete Handling of Service Options After=Submitted (see page 527).

            Otherwise, approvers *cannot* approve and reject individual service options, even if you configure your request approval email to support this feature.

    ■   Help ensure that the email is user friendly on all devices, including PDAs

    To achieve these goals, do *one* of the following, whichever is applicable for your implementation:

    ■   Customize your Notify Approver action to support PDA Approval (see page 542), by configuring the request approval email sent by this action

    ■   Customize your USM_Approval process definition to support PDA Approval (see page 550), by configuring the request approval email sent by this process definition

    By default, CA Service Catalog sends the request approval email automatically to approvers as soon as a user submits a request for approval. When you enable PDA approval, approvers on both office computers (laptops and desktops) and on PDAs receive *nearly* the same version of the email.

    You can optionally customize the content of this email at any time, whether you are using PDA approval or not. If you are using PDA approval, you can optionally review, modify, and test this email in actual requests to help ensure that it is user friendly on all devices, including PDAs.

9. Optionally customize the request email profile file (see page 555), to control the level of detail displayed in the request snapshot.

    The request snapshot is an optional summary of the details of the request that appears at the bottom of your request approval email. If you do not want to include the request snapshot or if you do not want to customize it, you can skip this step. The request snapshot applies to *both* CA Process Automation approval and CA Workflow approval.

10. Log in to CA Service Catalog from an office computer and change your default password, if you have not already done so. The default password is supplied either at installation time or when an administrator created or imported your user ID.

    This step is a prerequisite for all users who plan to approve and reject requests from a PDA.

11. Review how to approve and reject requests from a PDA (see page 557); share this information with the users to whom it applies.

Possible Limitations of PDA Approval

Organizations must be aware of these possible limitations while working with PDAs. If any of these possible limitations apply, your organization must decide how to handle them, and instruct administrators and users accordingly. If these limitations apply, they are most likely to affect older PDAs with older browsers.

■ PDA approvers may not be able to add comments and attachments to the request when approving or rejecting requests.

■ The browsers may not be able to render forms completely correctly. Consequently, the following limitations may exist when users (including approvers) access a form in a request through a PDA. Therefore, for complete support of forms in a request, you many need to access the request by logging in to CA Service Catalog from a laptop or desktop computer.

  – PDA users may be able to *view* most information on forms but may *not* be able to enter information, even for required fields. On an office computer, the attempt to approve or reject a request fails if the approver does not complete the required fields. However, on a PDA, the attempt to approve or reject a request may succeed *even when* the approver does not complete the required fields.

    Important! Instruct approvers to approve or reject a service using an office computer (desktop or laptop), *not* a PDA when they cannot enter information in the form fields.

    This limitation may apply to Form Designer forms but not to CA Workflow forms.

– Any hidden fields on the forms may appear as visible (*not* hidden). Therefore, as an administrator, verify that any information included in hidden fields is suitable for viewing by approvers or other users who can access the request using a PDA. Otherwise, remove the hidden fields from the forms.

This limitation may apply to Form Designer forms but not to CA Workflow forms.

– Text boxes, text areas, and select boxes (single select type) typically appear and function the same on both PDAs and office computers. However, other elements of a form (see page 281) (such as radio groups, field sets, and lookup fields), instead of displaying the *label* attribute, may display other data, including but not limited to, unique IDs, mapping labels, or some combination of such data.

This limitation may apply to both Form Designer forms and CA Workflow forms.

Therefore, if you as an administrator use such elements on forms, verify that all users can understand the meaning of the related information displayed in the request email on PDAs.

Set the Request Management Configuration Parameters

Setting the request management configuration parameters is a required task to enable PDA approval.

To set the request management configuration parameters

1. Click Service Builder, Request Management Configuration.

2. Set the following parameter:

    PDA Support: Enable

    Specify Yes to enable PDA support or No to disable PDA support.

    When you specify Yes, PDA users can click the links provided in PDA-compliant request approval emails to access the requests and approve or reject them.

    If you specify Yes, a message may appear warning you that support for forms limited when you enable PDA Approval. Review the limitations of PDA approval (see page 535) and answer the prompt to continue or cancel.

    When you specify No, the links for accessing the requests and approving or rejecting them do *not* appear in the request approval emails that PDA users receive. The absence of such links implicitly guides PDA users to use an office computer to view, approve, and reject requests.

3. Click Save.

Direct Email Approval

You can optionally set up the approval directly through email processes in your system so that assignees of requests pending action approve or reject the requested services and service options directly through email. The assignees approve or reject such requests by replying directly to an email from the Catalog system, without logging in to CA Service Catalog.

Using such a direct email approval method eliminates the network and geographic dependencies required for logging in to CA Service Catalog to approve and reject requests securely. However, the network and geographic dependencies required for running email still apply.

For ease of reference, this topic and related topics use these terms:

■ PDA Approval – for enabling users to approve and rejects requests using a PDA, either with or without logging in to CA Service Catalog.

■ Direct email approval – for approving requests directly through an email generated by the Catalog system, without logging in to CA Service Catalog. Users approve and fulfill requests by clicking the Approve or Reject link in the email.

■ Logged in approval – for approving requests by logging in to CA Service Catalog, using an email generated by the Catalog system. Users approve or reject requests by logging in to CA Service Catalog, displaying their own requests pending action, and acting on them.

Note: You can optionally implement *both* methods. However, as a best practice, for maximum efficiency of request processing, including maintenance and troubleshooting, we suggest that you use only one of these approval methods.

With or without direct email approval in use, CA Service Catalog records audit history, such as which user updated the statuses of a request, service, or service option.

Take special care when forwarding emails sent through direct email approval, as noted in the setup process (see page 537) for direct email approval.

How to Set Up Direct Email Approval

Important! This topic and its linked topics apply *only* if you use direct email approval (see page 537) (*not* logged in approval) as your method of PDA approval! If you use logged in approval, this topic and its linked topics do *not* apply.

To set up direct email approval, follow this process:

1. Review the comparison direct email approval and logged in approval (see page 538) and verify that you want to use direct email approval.

2. Meet the prerequisites (see page 540).

3. Do one of the following, whichever applies to your implementation:

   - Customize the request approval email sent by the notify approver action to support direct email approval (see page 547)

   - Customize the request approval email sent by the USM_Approval Process Definition to support direct email approval (see page 554)

4. Test the implementation to verify that the approvals and rejections are received and processed by the Catalog system as expected:

   a. Verify that requests approved using direct email approval move to the Approved status of the request life cycle, by default 800.

   b. Verify that requests rejected using direct email approval move to the Rejected status of the request life cycle, by default 600.

   c. If you have enabled approval by individual services and service options, verify that the approval and rejection actions occur as configured and as expected.

5. Instruct all approvers to take special care *not* to manually forward, auto-forward, or auto-delegate request approval emails to unauthorized users. This precaution applies especially to company officers who go on vacation and configure their email program to set up email delegates or to auto-forward their emails. Otherwise, unauthorized users may be able to view request details, which may be confidential. If you are unable to configure your email software this way, consider disabling direct email approval temporarily while you work to resolve any related issues.

   When you use direct email approval, the Catalog system maintain security as follows: If the email recipient forwards the email, the direct approval links do not work for any user who receives the forwarded email, unless that user also has both a valid email address specified in its user profile (see page 80) and a role (see page 89) in CA Service Catalog authorized to approve or reject that request, service, or service option.

Comparison of Direct Email Approval and Logged In Approval

The following table summarizes a comparison of direct email approval (see page 537) and logged in approval.

	Direct Email Approval	Logged In Approval
CA Process Automation	yes	yes
CA Workflow	*no	yes
IMAP Required	yes	no
Network dependency	for accessing email only	for accessing both email and CA Service Catalog
Supports custom statuses for requests	**yes	**yes

Supports approval and rejection of individual services	***yes	***yes
Supports approval and rejection of individual service options	***yes	***yes
Applicable to PDA	yes	yes

*If you are already using CA Workflow for both approval and fulfillment, but you want to add direct email approval, you can optionally install CA Process Automation and use it for direct email approval only, while optionally retaining CA Workflow for logged in approval. While we do not suggest using both direct email approval and logged in approval, this setup is possible.

**You must customize your request approval email to use your custom statuses. This email is sent by the Notify Approver action (see page 545) or sent by the USM_Approval process definition (see page 553), whichever is applicable for your implementation.

***You can customize your request approval email to support approval and rejection of individual services, if your request approval email is sent by the Notify Approver action (see page 545). Moreover, you can customize your request approval email to support approval and rejection of individual service options, if *both* of the following conditions are met:

- Your request approval email is sent by the Notify Approver action

- You are using a discrete request life cycle with approval set at the service option level (Allow Discrete Handling of Service Options After=Submitted (see page 527)

Meet the Prerequisites

Meeting these prerequisites is required to set up and use direct email approval.

Meet the prerequisites

1. Verify that your email server meets these prerequisite:

    ■   Your email server must support the IMAP protocol. CA Service Catalog provides a sample CA Process Automation approval process for using IMAP for direct email approval. This sample approval process requires that you set up an email account to which user's replies to requests pending action for approval are sent.

        See your CA Process Automation documentation for the IMAP configuration settings that you must use.

    ■   Your email server must be capable of efficiently processing the additional email volume generated by your using direct email approval.

    ■   Your email server must have the email account used by direct email approval; create an account, if necessary.

2. If you use CA Workflow for approving CA Service Catalog requests, and you are want to enable direct email approval, you must install CA Process Automation. Direct email approval requires CA Process Automation to function correctly.

3. Configure CA Process Automation to support the IMAP configuration. This configuration is set in the Mail Trigger Properties of the Configuration Browser in the CA Process Automation client. In these properties, do the following:

    –   In the Default process handler field, enter the following literal value: /CA SLCM/EmailApproval.

    –   In the User Name field, enter the email account (for example, the address of the account) used by direct email approval (created earlier in this topic)

        Record this value for use in a later step.

    For information about installing and configuring CA Process Automation, see your CA Process Automation documentation. The CA Process Automation installation media and documentation are supplied with the CA Service Catalog installation media.

4. Load all CA Process Automation content, if you have not already done so.

    Note: For details about loading this content, see the *Integration Guide*.

    Loading this content makes all processes active.

5. If you are using the Notify Approver - Email based approval action (or a custom version of it) for approving requests, set your send address in *only one* of the following; otherwise, skip this step.

    Set this address to be the same as the one specified for the User Name field of Mail Trigger Properties for CA Process Automation; you recorded this value in an earlier step.

    ■ If you want your request notification email address to be t*he same as* your email approval address, click Service Builder, Configuration, Request Management Configuration, and check the value of the parameters named Request Email: From Address.

    ■ If you want your request notification email address to be *different from* your email approval address, Click Administration, Tools, Events-Rules-Actions, open the Request Pending Action Change event, and open its When Status is Pending Approval rule.

        Verify that both the rule and its Notify Approver - Email based approval action are active.

        In the action, the From Email specifies the email sending address.

6. If you are using the USM_Approval process definition (or a custom version of it) for approving requests, set your send address in that process definition; otherwise, skip this step.

    Note: For details about USM_Approval and other predefined process definitions, see the CA Workflow chapter of the *Integration Guide*. For details about creating and customizing process definitions, see the CA Workflow IDE online help.

7. Verify that the Catalog system is not sending duplicate email notifications by submitting, approving, fulfilling, and completing requests.

8. Verify that all approvers in the Catalog system have a valid email address specified in their user profiles (see page 80). If necessary, edit the user (see page 86) to add the email address.

Customize Your Notify Approver Action to Support PDA Approval

If you use CA Process Automation for approving CA Service Catalog requests, you typically use the Notify Approver action (or a customized version of it) to send request approval emails, as explained in this topic. If you use CA Workflow for approving CA Service Catalog requests, you typically send request approval emails using *either* the Notify Approver action *or* the USM_Approval process definition (see page 553), but *not* both. If you use both to send request approval emails, approvers receive duplicate emails for the same requests.

To review and customize your Notify Approver action to support PDA approval

1. Click Administration, Tools, Events-Rules-Actions.

   The events appear.

2. Click the Request Pending Action Change event.

   The rules for that event appear.

3. Click the rule named When Status is Pending Approval (or your customized version of it) and verify that it is enabled.

4. Click the Notify Approver - Email based approval action (or your customized version of it) and verify that it is enabled.

5. Click the Edit button for this action. Review the specifications, especially for the required fields and the text in the Message field. Especially note the following updates for supporting PDA approval:

   That text is used to construct the request approval email and is divided into two sections separated by the $is_pda_enabled=$pda_enabled$$ marker.

   If PDA approval is *not* enabled in the Request Management configuration parameters (see page 536), the Catalog system uses the text *before* the $is_pda_enabled=$pda_enabled$$ marker to construct the request approval email. This text appears similar to the following:

   One or more items in the request $request_name$ ($request_id$) for $req_for_user_id$ for service $offering_name$ has been submitted to your approval queue. Please review this request by clicking here</br>

If PDA approval *is* enabled in the Request Management configuration parameters, the Catalog system uses the text *after* the $is_pda_enabled=$pda_enabled$$ marker to construct the request approval email. By default, this text appears similar to the following:

One or more items in the request $request_name$ ($request_id$) has been submitted to your approval queue. The summary of the request is included in this email.

Click links below to perform the approval action on the request.

 Approve Request without further review

Reject Request without further review

 Review the Details of the request in CA Service Catalog; afterwards you can approve or reject some or all of the requested items.

Please wait for the confirmation message in your browser after performing the approval action.

The default parameters in this text are interpreted at runtime as follows:

$REQUEST_APPROVE_DETAILS$

Provides the details link when PDA support is *not* enabled for the business unit in which the request is created and submitted. When the approver who receives the email clicks this link, the request UI opens and displays the request details.

$UPDATE_STATUS_LINK=HTTP|800|R$

Provides the link that the approver clicks to approve the entire request.

$UPDATE_STATUS_LINK=HTTP|600|R$

Provides the link that the approver clicks to reject the entire request.

These links enable logged in approval, for approving and rejecting entire requests *only*. These are the default links.

Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links, as explained later in this procedure.

6. If you *are* already using a custom version of the Notify Approver - Email based approval action, do the following; otherwise, skip this step and resume at the next step:

   a. Copy the updates in the predefined version to your custom version, especially the updates to the Message section text described in the previous steps.

   Note any customizations that you have previously made, and make them to the new text, if applicable.

   b. As a best practice, verify that the setting for Include Request Details field is Yes.

   c. Skip the next step and resume at Step 8.

Important! When you copy and modify the text from the predefined version to your custom version, do *not* delete any of the parameters explained earlier!

7. If you *are not* already using a custom version of the Notify Approver - Email based approval action, do the following; otherwise, skip this step and resume at the next step: Click Copy to copy and modify the action, so that you can start using a custom version.

    When prompted, specify a meaningful name for the new action, such as Notify Approver - Email based approval--custom BU version.

    You must copy and modify the action to support PDA approval, because the predefined versions of all actions permit only very limited editing.

    For details about the fields, see the Request Email fields available when you add a rule action (see page 32).

    Keep in mind the following:

    - In your custom text in the Message field, include all parameters (explained in a previous step) from the predefined text.

    - For best results, verify that the text in the Message field is user-friendly on both office computers and PDAs, because it is used at the beginning of the request approval.

    - Verify that the setting for Include Request Details field meets your needs. If you specify Yes, the Catalog system creates an HTML summary of the request and includes it in the request approval email. If you specify No, the request approval email does not include such a summary.

      As a best practice, specify Yes.

    - To use the '$' character in the message body, use the backslash (\) as an escape character, as follows: \$.

    - Do *not* delete any of the parameters explained earlier.

8. Select the predefined version of the Notify Approver - Email based approval action and click Disable to disable it (if it is enabled).

    Important! Disabling the predefined version of the action is very important to prevent duplication and potential conflicts with the custom version.

9. Do one of the following, depending on the method of PDA approval that you want to use:

    - Customize the request approval email sent by the Notify Approver action to use logged in approval (see page 545)

    - Customize the request approval email sent by the Notify Approver action to use direct email approval (see page 547)

Customize the Request Approval Email Sent by the Notify Approver Action to Use Logged In Approval

Configuring the approval method for your PDA is a required task in process of setting up PDA approval (see page 533). As part of that process, you complete the initial steps to customize your Notify Approver action to support PDA approval (see page 542). Next, you configure your implementation of PDA approval to use logged in approval, as explained in this topic.

To customize the request approval email sent by the Notify Approver action to use logged in approval

1. Do the following to configure the request approval email to include links for approving and rejecting individual services; otherwise skip this step.

   a. As a best practice, *replace* the default links for approving and rejecting entire requests only with the following links:

      $UPDATE_STATUS_LINK=HTTP|800|S$

      Provides the link that the approver clicks to approve an individual service.

      $UPDATE_STATUS_LINK=HTTP|600|S$

      Provides the link that the approver clicks to reject an individual service.

      When adding the lines, use the same format and control characters as the rest of the email, to provide a consistent look-and-feel, for example:

 Approve Service without further review

      As a best practice for ease of use for approvers, *replace* the existing approval links with the new links, rather than *adding* the new links to the existing ones. For maximum usability, write your request approval email to include approval and rejection links for *only one* of the following:

      ■ The entire request

      ■ Individual services in the request

      ■ Individual service options in the services in the request

      You can optionally specify combinations of these links in the email. However, approvers may become confused by the overlapping options in the email text.

      Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links, as explained later in this procedure.

a. Update the descriptive text *above* each new link to match the change in the link.

    For example, update the text above the approval link to be similar to the following: "Click the link to approve the *service*..."

The resulting request approval email that users receive contains one highlighted service for them to approve or reject. If a request includes multiple services, users receive one request approval email for every service that they are assigned to approve or reject.

2. Do the following to configure the request approval email to include links for approving and rejecting individual service options; otherwise skip this step.

    a. Verify that you are using a discrete request lifecycle with approval set at the service option level (Allow Discrete Handling of Service Options After=Submitted (see page 527). Otherwise, approvers *cannot* approve and reject individual service options, even if you configure your request approval email to support this feature.

    b. As a best practice, *replace* the default links for approving and rejecting entire requests *only* with the following links:

        $UPDATE_STATUS_LINK=HTTP|800|SO$

        Provides the link that the approver clicks to approve an individual service option.

        $UPDATE_STATUS_LINK=HTTP|600|SO$

        Provides the link that the approver clicks to reject an individual service option.

        When adding the lines, use the same format and control characters as the rest of the email, to provide a consistent look-and-feel, for example:

        `<br> <a href='$UPDATE_STATUS_LINK=HTTP|800|S$'>Approve Service Option</a> without further review<br>`

        As a best practice for ease of use for approvers, *replace* the existing approval links with the new links, rather than *adding* the new links to the existing ones. For maximum usability, write your request approval email to include approval and rejection links for *only one* of the following:

        ■ The entire request

        ■ Individual services in the request

        ■ Individual service options in the services in the request

        You can optionally specify combinations of these links in the email. However, approvers may become confused by the overlapping options in the email text.

        Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links, as explained later in this procedure.

a. Update the descriptive text *above* each new link to match the change in the link.

For example, update the text above the approval link to be similar to the following: "Click the link to approve the following *service option*..."

The resulting request approval email that users receive contains one highlighted service option for them to approve or reject. If a request includes one or more services with multiple service options, users receive one request approval email for every service option that they are assigned to approve or reject.

3. Skip this step if your organization is using the default status values: 800 for Approved and 600 for Rejected.

If your organization is custom status values instead, update the status values in all links in the request approval email to match your custom values for Approved and Rejected, as defined in your requestshared.xml file.

Note: For information about viewing and editing the requestshared.xml file, see the *Implementation Guide*. The range for custom status values for Approved is 800-999; for Rejected, 600-799.

You have reviewed and customized the request approval email in your custom version of the Notify Approver - Email based approval action.

Important! This note applies *only* if you are using CA Workflow for approving and rejecting requests. After you complete this procedure, do the following: Verify that the usm_approval.xml file uses the same links as the request approval email in your custom version of the Notify Approver - Email based approval action. If necessary, update the links in that file to match the ones in the request approval email.

Customize the Request Approval Email Sent by the Notify Approver Action to Use Direct Email Approval

Configuring the approval method for your PDA is a required task in process of setting up PDA approval (see page 533). After you compete the initial steps to customize your Notify Approver action to support PDA approval (see page 542), continue here to configure your implementation of PDA approval to use direct email approval.

1. Do the following to configure the request approval email to include links for approving and rejecting entire requests, using direct email approval; otherwise, skip this step.

   Replace the default approval and rejection links with the following links, which specify direct email approval rather than logged in approval:

   $UPDATE_STATUS_LINK=MAIL_TO|800|R|Subject|Body|$

   Provides the link that the approver clicks to approve the entire request.

   $UPDATE_STATUS_LINK=MAIL_TO|600|R|Subject|Body|$

   Provides the link that the approver clicks to reject the entire request.

   Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links, as explained later in this procedure.

2. Do the following to configure the request approval email to include links for approving and rejecting individual services; otherwise, skip this step.

   a. As a best practice, *replace* the links for approving and rejecting entire requests with the following links:

      $UPDATE_STATUS_LINK=MAIL_TO|800|S|Subject|Body|$

      Provides the link that the approver clicks to approve an individual service.

      $UPDATE_STATUS_LINK=MAIL_TO|600|S|Subject|Body|$

      Provides the link that the approver clicks to reject an individual service.

      When adding the lines, use the same format and control characters as the rest of the email, to provide a consistent look-and-feel, for example:

 Approve Service without further review

      As a best practice for ease of use for approvers, *replace* the existing approval links with the new links, rather than *adding* the new links to the existing ones. For maximum usability, a request approval email includes approval and rejection links for *only one* of the following:

      ■ The entire request

      ■ Individual services in the request

      ■ Individual service options in the services in the request

      You can optionally specify combinations of these links in the email. However, approvers may become confused by the overlapping options in the email text.

      Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links, as explained later in this procedure.

a. Update the descriptive text *above* each new link to match the change in the link.

For example, update the text above the approval link to be similar to the following: "Click the link to approve the following *service*..."

The resulting request approval email that users receive contains one highlighted service for them to approve or reject. If a request includes multiple services, users receive one request approval email for every service that they are assigned to approve or reject.

3. Do the following to configure the request approval email to include links for approving and rejecting individual service options; otherwise skip this step.

a. Verify that you are using a discrete request life cycle with approval set at the service option level (Allow Discrete Handling of Service Options After=Submitted (see page 527). Otherwise, approvers *cannot* approve and reject individual service options, even if you configure your request approval email to support this feature.

b. As a best practice, *replace* the links for approving and rejecting entire requests with the following links:

$UPDATE_STATUS_LINK=MAIL_TO|800|SO|Subject|Body|$

Provides the link that the approver clicks to approve an individual service option.

$UPDATE_STATUS_LINK=MAIL_TO|600|SO|Subject|Body|$

Provides the link that the approver clicks to reject an individual service option.

When adding the lines, use the same format and control characters as the rest of the email, to provide a consistent look-and-feel, for example:

 Approve Service Option without further review

As a best practice for ease of use for approvers, *replace* the existing approval links with the new links, rather than *adding* the new links to the existing ones. For maximum usability, write your request approval email to include approval and rejection links for *only one* of the following:

■ The entire request

■ Individual services in the request

■ Individual service options in the services in the request

You can optionally specify combinations of these links in the email. However, approvers may become confused by the overlapping options in the email text.

Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links, as explained later in this procedure.

a. Update the descriptive text *above* each new link to match the change in the link.

    For example, update the text above the approval link to be similar to the following: "Click the link to approve the following *service option*..."

The resulting request approval email that users receive contains one highlighted service option for them to approve or reject. If a request includes one or more services with multiple service options, users receive one request approval email for every service option that they are assigned to approve or reject.

4. Skip this step if your organization is using the default status values: 800 for Approved and 600 for Rejected.

    If your organization is custom status values instead, update the status values in all links in the request approval email to match your custom values for Approved and Rejected, as defined in your requestshared.xml file.

    Note: For information about viewing and editing the requestshared.xml file, see the *Implementation Guide*. The range for custom status values for Approved is 800-999; for Rejected, 600-799.

You have reviewed and customized the request approval email in your custom version of the Notify Approver - Email based approval action.

Important! This note applies *only* if you are using CA Workflow for approving and rejecting requests. After completing this procedure, verify that the usm_approval.xml file uses the same links as the request approval email. If necessary, update the links in that file to match the ones in the request approval email.

Customize Your USM_Approval Process Definition to Support PDA Approval

If you use CA Workflow for approving CA Service Catalog requests, you typically send request approval emails using *either* your Notify Approver action (see page 545) *or* your USM_Approval process definition, but *not* both. If you use both to send request approval emails, approvers receive duplicate emails for the same requests.

Note: For details about USM_Approval and other predefined process definitions, see the CA Workflow chapter of the *Integration Guide*. For details about creating and customizing process definitions, see the CA Workflow IDE online help.

To review and customize your USM_Approval process definition to support PDA approval

1. Select Administration, Tools, Links on the CA Service Catalog GUI.

    The page displays links to several software applications and tools.

2. Click the tool named CA Workflow IDE and log in to it.

    The CA Workflow IDE opens.

3. Locate and open the USM_Approval process definition.

4. Locate and open the First Level Approval sub-activity.

5. Locate and open the Email Approvers activity.

6. Go to the Workitem tab, under the Input Parameters section, and open the emailMessage parameter.

   The default text follows:

   concat($EmailFontStart,"One or more items in the request (",$RequestID,") for
   ",$UserFirstName," ",$UserLastName," for service ",$OfferingName," has
   been submitted to your approval queue.   Please review this request by clicking <a href=
   '$REQUEST_APPROVE_DETAILS$'>here.",$EmailFontEnd,"$is_pda
   _enabled=$pda_enabled$$",$EmailFontStart,"One or more items in the request
   (",$RequestID,") for ",$UserFirstName," ",$UserLastName," for service
   ",$OfferingName," has been submitted to your approval queue. The summary of the request is
   included in this email.

 <a href=
   '$UPDATE_STATUS_LINK=HTTP|800|R$'>Approve Request without further
   review

 Reject
   Request   without further review

 <a href=
   '$REQUEST_APPROVE_DETAILS$'>Review the Details.
 of the request
   in CA Service Catalog; afterwards you can approve or reject some or all of the requested items.

 Please wait for the confirmation message in your browser after performing the approval
   action.",$EmailFontEnd)

   This text is used to construct the request approval email and is divided into two sections separated by the $is_pda_enabled=$pda_enabled$$ marker, as follows:

   ■ If PDA approval is not enabled in the Request Management configuration parameters (see page 536), the Catalog system uses the text *before* the $is_pda_enabled=$pda_enabled$$ marker to construct the request approval email.

   ■ If PDA approval *is* enabled in the Request Management configuration parameters, the Catalog system uses the text *after* the $is_pda_enabled=$pda_enabled$$ marker to construct the request approval email.

   The default parameters in this text are interpreted at runtime. The following parameters may require explanation:

   $OfferingName

   Specifies the name of the service requested.

   Note: *Offering*, an older term for *service*, is retained here for compatibility.

   $REQUEST_APPROVE_DETAILS$

   Provides the details link for which the request is created and submitted. When the approver who receives the email clicks this link, the request UI opens and displays the request details.

$UPDATE_STATUS_LINK=HTTP|800|R$

Provides the link in the email that the approver clicks to approve the entire request.

$UPDATE_STATUS_LINK=HTTP|600|R$

Provides the link in the email that the approver clicks to reject the entire request.

Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links.

7. Do one of the following, whichever applies for your implementation:

   ■ If you are using the predefined version of the USM_Approval process definition, optionally update the process definition as explained in this procedure and linked procedures.

   ■ If you are *already* using a custom approval process definition (for example, a copied and modified version of the USM_Approval process definition), copy the updates in the predefined version to your custom version, especially message text in the emailMessage parameter described in the previous steps.

   In that case, be careful to retain any existing customizations in your custom approval process definition.

   Important! When you copy or modify the message text from the emailMessage parameter of the predefined version to your custom version, do *not* delete any of the parameters explained earlier!

   ■ If you want to begin using a custom version of the USM_Approval process definition, do the following:

     a. Copy the process definition according to the instructions in the CA Workflow IDE online help.

     b. Modify your custom version as explained in the remaining steps in this procedure and linked procedures.

     c. Disable the predefined version of the USM_Approval process definition according to the instructions in the CA Workflow IDE online help.

     Important! Disabling the predefined version of the action is very important to prevent duplication and potential conflicts with the custom version.

8. Do one of the following, depending on the method of PDA approval that you want to use:

   ■

   ■

Customize the Request Approval Email Sent by the USM_Approval Process Definition to Use Logged In Approval

Configuring the approval method for your PDA is a required task in process of setting up PDA approval (see page 533). After you compete the initial steps to customize your USM_Approval process definition to support PDA approval (see page 550), do the following: Configure your implementation of PDA approval to use logged in approval, as explained in this topic.

Note: When you use your USM_Approval process definition to send the request approval email, you can use the default email to support logged in approval, for approving and rejecting entire requests *only*. You *cannot* customize the email to support approving and rejecting individual services or service options.

To customize the request approval email sent by the USM_Approval Process Definition to use direct email approval

1. Verify that your request approval email includes the default links described the initial steps to customize your USM_Approval process definition to support PDA approval (see page 550). If you or another administrator changed the default links, update them to match the default specifications.

   As noted earlier, these default links support logged in approval for the entire request *only.*

2. Skip this step if your organization is using the default status values: 800 for Approved and 600 for Rejected.

   If your organization is custom status values instead, update the status values in all links in the request approval email to match your custom values for Approved and Rejected, as defined in your requestshared.xml file.

   Note: For information about viewing and editing the requestshared.xml file, see the *Implementation Guide*. The range for custom status values for Approved is 800-999; for Rejected, 600-799.

You have customized the request approval email sent by the USM_Approval process definition (or your customized version of it) to support logged in approval.

Important! After completing this procedure, verify that the usm_approval.xml file uses the same links as the request approval email. If necessary, update the links in that file to match the ones in the request approval email.

Customize the Request Approval Email Sent by the USM_Approval Process Definition to Use Direct Email Approval

Configuring the approval method for your PDA is a required task in process of setting up PDA approval (see page 533). After you compete the initial steps to customize your USM_Approval process definition to support PDA approval (see page 550), do the following: Configure your implementation of PDA approval to use direct email approval, as explained in this topic.

Note: When you use your USM_Approval process definition to send the request approval email, you can customize the email to support direct email approval, for approving and rejecting entire requests only. You cannot customize the email to support approving and rejecting individual services or service options.

To customize the request approval email sent by the USM_Approval Process Definition to use direct email approval

1. Replace the default approval and rejection links with the following links, which specify direct email approval rather than logged in approval:

   $UPDATE_STATUS_LINK=MAIL_TO|800|R|Subject|Body|$

   Provides the link that the approver clicks to approve the entire request.

   $UPDATE_STATUS_LINK=MAIL_TO|600|R|Subject|Body|$

   Provides the link that the approver clicks to reject the entire request.

   Important! The status values 800 and 600 are the default values for Approved and Rejected, respectively. If you are using custom status values for Approved and Rejected, replace the default values with your custom values in these links, as explained later in this procedure.

2. Skip this step if your organization is using the default status values: 800 for Approved and 600 for Rejected.

   If your organization is custom status values instead, update the status values in all links in the request approval email to match your custom values for Approved and Rejected, as defined in your requestshared.xml file.

   Note: For information about viewing and editing the requestshared.xml file, see the *Implementation Guide*. The range for custom status values for Approved is 800-999; for Rejected, 600-799.

You have customized the request approval email sent by the USM_Approval process definition (or your customized version of it) to support direct email approval.

Important! After completing this procedure, verify that the usm_approval.xml file uses the same links as the request approval email. If necessary, update the links in that file to match the ones in the request approval email.

Optionally Customize the Request Email Profile File

You can optionally customize certain lines of the requestemailprofile.xsl file, to control the level of detail displayed in the request snapshot. The request snapshot is an optional summary of the details of the request that appears at the bottom of your request approval email. If you do not want to include the request snapshot or if you do not want to customize it, you can skip this step. The request snapshot, if used, applies to both CA Process Automation approval) and CA Workflow approval. When you set up your approval process, you decide whether to include the request snapshot.

Note: If you did not include the request snapshot originally when you set up your approval process, but you want to include the request snapshot now, see the instructions for configuring your approval process (either CA Process Automation or CA Workflow) in the *Integration Guide*.

Important! If a parameter in this file is not described in this topic, do *not* modify the parameter!

The requestemailprofile.xsl file is located in the %USM_HOME%/view/webapps/usm/explorer/request folder.

You can optionally modify the following parameters in the file, by setting its value to true or false. By default, all values are set to true.

■ The variable that controls whether the request snapshot displays the information in the Requested Services table.

<xsl:variable name="reqservices" select="true()" />

The Requested Services table includes the following:

■ The variable that controls whether the request snapshot displays the names and descriptions of the service options included in each service in the request:

<xsl:variable name="infosog" select="true()" />

Use this variable to either show both the names and descriptions or hide both of them.

■ The variable that controls whether the request snapshot displays the descriptions of the service options included in each service in the request.

<xsl:variable name="descriptionsog" select="true()" />

Use this variable to show or hide only the descriptions.

■ The variable that controls whether the request snapshot displays a summary of the fields included in each form for each service in the request:

<xsl:variable name="formsinfo" select="true()" />

■   The variable that controls whether the request snapshot displays the information in the General Information table:

<xsl:variable name="infosectionsView" select="true()" />

The General Information table includes the following:

■   The variable that controls whether the request snapshot displays the Request Name field in the General Information table:

   <xsl:variable name=" reqname" select="true()" />

■   The variable that controls whether the request snapshot displays the Request ID field in the General Information table:

   <xsl:variable name=" reqid" select="true()" />

■   The variable that controls whether the request snapshot displays the Requested for field in the General Information table:

   <xsl:variable name=" reqfor" select="true()" />

■   The variable that controls whether the request snapshot displays the Request Status field in the General Information table:

   <xsl:variable name=" reqstat" select="true()" />

■   The variable that controls whether the request snapshot displays the Requested by field in the General Information table:

   <xsl:variable name=" reqby" select="true()" />

■   The variable that controls whether the request snapshot displays the Date Created field in the General Information table:

   <xsl:variable name=" datecreated" select="true()" />

■   The variable that controls whether the request snapshot displays the Priority field in the General Information table:

   <xsl:variable name=" priority" select="true()" />

■   The variable that controls whether the request snapshot displays the Date Required field in the General Information table:

   <xsl:variable name=" daterequired" select="true()" />

■   The variable that controls whether the request snapshot displays the Last Modified field in the General Information table:

   <xsl:variable name=" lastmodified" select="true()" />

Approve and Reject Requests Pending Action from a PDA

After the CA Service Catalog administrator has finished setting up CA Service Catalog for PDA approval, approvers can approve and reject requests from their PDAs. Depending on the setup specified by the administrator, you may be able to approve or reject any or all of the following in each request pending action you receive:

- Entire requests

- Individual services

- Individual service options

To approve and reject requests pending action from a PDA

1. Using your PDA, open the request approval email as soon as possible after you receive it.

2. Review the services and service options in the request and the assigned users. Decide whether to approve or reject the request, services, or service options assigned to you. Before doing so, you can optionally view further details in the request.

   Your organization can use policies to manage requests (see page 451) and thereby direct that *only* the users assigned by the policy (rather than any other users) approve or reject requests, services, and service options. In such cases, approve or reject only the individual services and service options explicitly assigned to you, even if you have proxy authority to approve or reject other users' requests pending action (see page 610).

   Do not approve or reject an entire request pending action unless all services and all service options in the request are explicitly assigned to you.

3. Click the link for the action you want to perform.

   Depending on how the administrator set up the request approval email for PDA approval, you may see only a subset of these options, for example, only the options for approving or rejecting entire requests or only the options for approving or rejecting individual service options.

   Approve Request without further review

   This option approves all the requests pending action *that are assigned to you*.

   In addition, this option approves *all other* requests pending action remaining for the request, if your role permits you to do so, even if you are not the assignee. For example, if your role is Service Delivery administrator (named spadmin by default), you can approve all requests pending actions for all requests.

   Approve Service without further review

   This option approves the highlighted service in the request pending action; the service assigned to you is highlighted in the request summary.

Approve Service Option without further review

This option approves the highlighted service option in the request pending action; the service option assigned to you is highlighted in the request summary.

Reject Request without further review

This option rejects all the requests pending action *that are assigned to you*.

In addition, this option rejects *all other* requests pending action remaining for the request, if your role permits you to do so, even if you are not the assignee. For example, if your role is Service Delivery administrator (named spadmin by default), you can reject all requests pending actions for all requests.

Reject Service without further review

This option rejects the highlighted service in the request pending action; the service assigned to you is highlighted in the request summary.

Reject Service Option without further review

This option rejects the highlighted service option in the request pending action; the service option assigned to you is highlighted in the request summary.

Review the details

This option displays the CA Service Catalog login page, so you can log in and review the additional information about the request to help decide whether to approve or reject it.

If you chose to log in to CA Service Catalog, complete the remaining steps.

4. Enter your user ID and password to log in to CA Service Catalog when prompted, if applicable, as follows:

If your implementation uses direct email approval (see page 537), you are not prompted or required to log in to CA Service Catalog to complete the action.

■ If your user ID is pre-populated, enter your password.

■ If the approval email is sent to a user group or to multiple individual users, then the user ID is not pre-populated, and you must enter both your user ID and password.

If your implementation is using logged in approval, each option prompts and requires you to log in to CA Service Catalog. After your login credentials are validated, the action is completed, and a confirmation message displays.

5. Do the following if you require more information than you can view using the PDA, or if you cannot complete required fields on a form. These steps also apply if you are unable to approve or reject the request for any reason.

   Do *not* use your PDA to approve or reject any request, service, or service option that brings concerns or questions to mind, especially for services that include forms requiring your input. At publication time, browsers for some PDAs (especially older ones) display only limited information (see page 535) about the request.

   a. Cancel your action to use the PDA to approve or reject any service or request of concern.

   b. Log in to CA Service Catalog from a laptop or desktop computer.

   c. At that computer, review the complete request details and approve or reject (see page 608) the service option, service, or request of concern.

   d. If applicable, enter any input required on forms.

6. If applicable, click Logout to exit CA Service Catalog from your PDA.

Chapter 14: Using CA Service Catalog for Catalog Users

This section contains the following topics:

Introduction for Catalog Users

All CA Service Catalog users can browse the catalog and request services and service options. They can also view recent requests, all open requests, completed requests and requests pending action on their part as well as look up.

For users in the Catalog User role, the first page after logon is the Requests page. For users in other roles, the request page is accessible from the Home tab. From the Requests page, a user can:

■ Browse the catalog by drilling down catalog folders to view services offered

■ Search the catalog for specific entries

■ View services designated as featured items

■ View recent requests

■ View the shopping cart of selected services

■ View the list of open requests

■ View the list of completed requests

■ View the list of requests pending action by the user

■ View all requested or subscribed items

For Catalog Users, other capabilities on the header and footer portions of the products page include:

■ Welcome bar which displays the user name, a log out link, the role and business unit specified during login

■ Requests link to the Request page

■ Dashboards link to the Dashboards page

■ Profile link to display the logged in user's information and allow a password change

■ Help link to the product documentation

■ About link to the about page

Log In

You log in to CA Service Catalog so that you can access it and use its features and functions.

To log in

1. Use your browser to navigate to the URL, provided by your system administrator, in the format http://*computer-name*:*port-number*.

    If CA Service Catalog is configured to use single signon, you are automatically authenticated and the initial page appears. Otherwise the Log In page appears.

2. If the Log In page appears, enter your assigned User Name and Password.

3. If you only have one role assigned or if you want to use your default role, click Log In.

    Your login credentials are authenticated and the initial page appears.

4. If you have multiple roles, click Advanced.

    The Business Unit ID field appears.

5. Enter the business unit login ID for the role that you want to use.

    Your login credentials are authenticated and the initial page appears.

Request a Service

Requesting a service from the catalog is similar to shopping from a catalog. This experience is familiar to anyone who has shopped online.

To request a service from the catalog

1. Browse or search the catalog to find the service you are interested in requesting.

2. Display the options for the service and select the options you want.

3. Add the selected service options to your cart.

   Continue shopping until you want to check out.

4. View your cart to check out or check out from the Add To Cart page.

5. View the My Selections section to confirm your selections and remove any item that you do not want to keep.

6. View the General Information section for additional read-only information about your request.

7. Confirm the information in your cart and submit your cart.

Depending on your service provider's policies regarding the items you have requested, your request has to be approved. After all the services you have requested are approved, the items you have requested needs to be fulfilled. After all the items have gone through the fulfillment phase, your request is considered *completed*.

564   Administration Guide

Add a Catalog Service to your Cart

The entries in the service catalog that you view are managed by your service provider's catalog administrator. The catalog represents the services made available to you by your service provider and can include choices appropriate for those services. For example, if the catalog has a service named *Email* it can have service options named *Small mailbox size* and *Large mailbox size,* and you select only one of them. Similarly, if the catalog can have a service named *Desktop* that requires you to select either *Standard* or *Deluxe* but also enables you to select one or more software options.

Use one of the following methods to find a service in the catalog:

- Browse the catalog

- Examine Featured Services

- Search the catalog

The catalog is organized by a category hierarchy created by your catalog administrator. To browse the catalog, click one of the categories or sub-categories shown in the Browse section of the Requests page. As you drill down into a category, the sub-categories contained in that category appear in the Browse section. The Catalog Services section displays the services in the selected category. Services can be contained in any category or in no category (shown by clicking the Not Categorized link in the initial Browse window). As you drill into the category hierarchy, you can see that the *breadcrumbs* shown at the top of the window reflect your path into the hierarchy and provide a way for you to navigate back to previous levels in the hierarchy. After you have selected a category that contains services, they appear in the Catalog Services section of the window. To further examine or select a service, click its name in the Catalog Services section of the window.

Your catalog administrator can choose to feature certain services from the catalog when you select a category or other service. If so, you can view the Featured Services section of the window appear when you select a related category or service. You can further examine or select a featured service by clicking its name in the Featured Service section of the window.

You can also use the Search In area of the window to search the entire catalog or selected category for services whose identifying values such as name or description match the text you enter. When you enter selection criteria and click Go, the Search Results list of services appears. You can further examine or select a service by clicking its name in the Search Results list.

After you have selected a service by clicking its name, the Service Options page appears. This page displays all the service options available with the service. Related service options are organized into a service option group. A service can include several service option groups, each displayed in its own collapsible section of the Service Options page. Some service option groups require a choice of one or more options (presented using a radio button). Other service option groups allow you to select one or more check boxes with the option of not selecting any check boxes. The third type of service option group automatically includes the service options displayed and you cannot *unselect* the service options.

Example

The Desktop service may include three service option groups: Desktop Type, Required Software, and Optional Software. The Desktop Type service option group may present the choices Standard and Deluxe, one of which you can select using a radio button. The Required Software service option group might include Microsoft Windows XP Pro, or Microsoft Office with no radio button or check box, meaning that the service includes these options automatically. The Optional Software service option group may include Microsoft Visio and Adobe Photoshop with check boxes, meaning that you can optionally select no, one, or multiple service options.

Depending on your role and the options selected by your catalog administrator, you may see a *See fulfillment details* link for each service option. If you click this link you can view these statistics for the service option:

■ The average time from submission to fulfillment in days and hh:mm.

■ The estimated time to fulfillment once the request is submitted in days and hh:mm.

■ The number of times you have requested this service option.

When you have selected all the service options you need for the service, click either Add to Cart to add this service to your shopping cart and continue shopping, or Add to Cart and Check Out to begin the check out process.

Note: Your catalog administrator may configure each request to allow only one service per cart, in which case you cannot view the Add to Cart button, only the Add to Cart and Check Out button.

If the selected service requires more information, one or more forms appear after you choose the service. Complete the required fields on the forms before you check out. For information about forms, see the chapter "Using Service Builder."

Check the Status of a Request

Monitoring the status of a request can help you understand where your request is in the request life cycle. Status values appear for the entire request and for each service and service option in the request. To better understand statuses, review the numbers, names, and meanings of default status values (see page 572).

The request status can be seen in the Status column of any request list, such as the My Recent Requests list, the Open Requests list, the Completed Requests list, the Pending Action list or the Request Search Results list. The request status can also be seen when viewing the details of a request.

To view the request status history, click the request status link. The Request Status History window appears. The Request Status History displays the request status, pending action status, the user ID of the user who set the status, and the date and time the status was set, shown in reverse chronological order.

Note: If an automated approval or fulfillment process set the status, the Updated By column displays *Catalog System*. If the request is pending an action by one or more users or groups, you can view the IDs of those users or groups for the most recent status in the list.

The Request Status History window contains the following main parts:

Status Summary

This summary appears in the top row under the Request Status column. This summary is an aggregate of the statuses of the services and service options in the request. If all service options in the request or service have the same status, the request or service status is that same status. A sample status summary is the following: Pending Fulfillment by spadmin/usr4/usr3/usr2. This sample summary indicates that the request is pending fulfillment by the four users named in the summary line.

Request Status

This column records all individual status changes for request, for example, Pending Approval, Pending Fulfillment, and so forth.

A summary of request status appears in the top row under the Request Status column. This summary is an aggregate of the statuses of the services and service options in the request. If all service options in the request or service have the same status, the request or service status is that same status. A sample status summary is the following:

Pending Fulfillment by spadmin/usr4/usr3/usr2

This sample summary indicates that the request is pending fulfillment by the four users named in the summary line.

Pending Action Status

This column records all individual actions performed on each service and service option in the request at different stages of request cycle, for example: assigned to user33, taken by spadmin, returned by spadmin, delegated to user55, transferred to user66, approved by spadmin, and so forth.

This column also records status changes for the service option or service as different users perform actions on it, for example: pending approval, approved, and pending fulfillment.

The date and time of each status change or action is also displayed.

To view the status history for a service option, first display the details of a request, then click the status link for a service option to display the Request Item Status History window. The display is similar to that shown in the Request Status History window but you can view more detailed information for the service option.

How to Check Out your Cart

To check out, click the Cart icon or link. The Cart window appears. Note that you may have navigated to the Cart window by clicking Add to Cart and Check Out when selecting a service. After you submit your cart, the information about your cart becomes a request so *cart* and *request* functionality is very similar.

The Cart window offers several choices:

■　View the requested services and service options (by clicking Show Details)

■　Delete a requested service from the cart

■　Alter several sections of additional information about the request

■　Display and add Notes

■　Display and add Attachments

■　Empty the cart

■　Save any changes to the cart without submitting the cart as a request

■　Save any changes to the cart and submit the cart as a request

■　Save the cart as an unsubmitted request so that you can create a new, separate cart (this option may not be available for all user roles depending on how the catalog administrator has configured your catalog)

View and Change Requested Services

The My Selections section of the cart or request displays the services and service options you have selected. Clicking Show Details displays all the service options you have selected while clicking Hide Details hides the service options and displays only the services. If you were asked to enter service-specific information when you selected the service, related forms appear, enabling you to verify and update the information you entered earlier.

The columns shown in the list of services included in the cart or request include the following:

■ Service - The name of the service or service option requested.

■ Status - This column is shown in a request but not a cart and indicates the current status of the service or service option.

Depending on your role and how your catalog administrator has configured the catalog, you may see these additional columns:

■ Amount - This column (if used) displays the price of the service option.

■ Cost/Other Info – This column displays information about the billing cycle, related CA Service Desk Manager tickets (if applicable), and price-related calculations.

You can remove a requested service from your cart or request by clicking the delete trash can icon in the Action column for the service you want to remove from your cart or request.

Note: If you have selected incorrect service options (like you selected the Standard desktop when you meant to select the Deluxe desktop) you must delete the request from the cart and re-select it from the catalog with the correct options chosen.

To save the changes to the requested services, click the Save Changes button.

View and Change Additional Information

There are several sections of additional information which appear if your catalog administrator has configured your catalog to display them. The information in these sections is for use by your service provider when approving and fulfilling the requested services.

One important field in the General Information section is the request Name. This field is pre-populated with the name of the first service you added to your cart but can be changed by you. This field is what appears in various request lists so you must be sure this field is set to a meaningful value to help you identify this request from other requests you can view in a list.

By default, the Requested For field is set to your user ID. If your role permits, you can change the Requested For field to be a different user or account. To change the Requested For field, click the adjacent magnifying glass icon to display the Search window. On the Search window you can search and select the user or account for which this request is being made.

To save the changes to the additional information sections, click the Save Changes button.

View and Change Notes

The Notes section of the cart or request can be used to view notes or add notes in rich text format.

Note: By default, the notes for the request are shown. Click the Click to Hide arrow to collapse the Notes section. Click the Click to Show arrow to view the notes for your request. The Notes header section indicates how many notes there are for the request.

To add a note to the request, click the Add button on the Notes section. If your role is permitted to add notes at the service or service option level, you can view an Add Note icon in the Action column. The Add Note window appears. Enter the text of the note. You can use any of the rich text options available on the Add Note window to enhance the appearance of your note. Click OK to save the note.

The note appears in the list of notes for the request. Notes cannot be deleted or altered.

View and Change Attachments

You can use the Attachments section of the cart or request to view associated files or add associated files to the request.

To view the attachments for your request, click the Show arrow. The Attachments header section indicates how many attachments there are for the request. Click the Click to Hide arrow to collapse the Attachments section.

To add a file to the request as an attachment, click the Add button on the Attachments section. The Add Attachment window appears. Enter the following fields and click OK:

- File - Type in the file path and name or click the Browse button to navigate to and select the file to be attached.

- Display Name - The name to display in the Attachments list for this attached file.

- Description - The description to display in the Attachments list for this attached file.

Clicking OK uploads a copy of the file to the file system on the server to which you are connected. The attachment appears in the attachments list. You can edit the display name or description of the attachment by clicking the Edit pencil icon in the Actions column. You can delete the attached file on the server by clicking the Delete trash can icon in the Actions column. You can view the attached file by clicking the Display Name value.

Empty the Cart

To clear the entire contents of the cart including any selected services, additional information, notes and attachments, click the Empty Cart button. You are prompted with a confirmation message and if confirmed, the cart is emptied and you are returned to the Requests window.

Save the Cart and Continue to Shop

If you have changed something about the cart and want to save your cart for continued shopping and later check out, click the Save Changes button. To continue shopping, click the Home link to return to the Requests window where you can browse the catalog for services.

Save the Cart and Check Out

After you have verified your cart and made any necessary changes, you must submit the cart for approval and fulfillment. To save any changes and submit your cart, click the Save and Submit Cart button. The Requests page appears with a confirmation message which includes the request ID number assigned to your submitted cart. Your request is shown in the list in the My Recent Requests area of the page and the View Cart link in the Request Lookup area of the page displays 0 services in your cart as your cart is now empty. Your request can now start the request life cycle.

Save the Cart as a Request

If your catalog administrator has enabled this feature, you can view the Save as Request link. This link allows you to save your cart as a request without submitting it. The request can have a status of Not Submitted until you edit and submit it. This feature allows you to use your cart to create multiple requests and submit any or all of the requests at a later time. Without this option enabled you must submit your cart before you can create another cart. This means you can effectively have only one unsubmitted request (your cart) being constructed at a time.

Names and Meanings of Status Values

When you view a request, the names of the current status appears in the Status column, Status drop-down list, or another location. When you click the list, the other available statuses appear; you can click one of them to move the request to a new status, such as Approved, Cancelled, or Fulfilled.

The following table shows the number, name, and meaning of the default statuses supplied with CA Service Accounting and CA Service Catalog. The number of a status does not appear in the drop-down list but is used in the requeststatus.xml file, which contains specifications for all default statuses and customized statuses (if any). By editing the requeststatus.xml file, you can optionally create new statuses and customize the statuses that appear in the drop-down list; for details, see the appendix "Customizing" in the *Implementation Guide*.

Some text in this table refers to the two request fulfillment models: simple and complex. In certain situations, the request flow differs for each model; such differences, when applicable, are noted in the following table. For details about the fulfillment models and the process definitions, see the chapter "Integrating with CA Workflow" in the *Integration Guide*.

Number	Name	Meaning
2	Completed	Catalog system-assigned status. When a request finishes with approval and fulfillment cycles, it reaches a Completed status. Once completed, the underlying request subscription reaches an active subscription status.
3	Pending Cancellation	Applies only when CA Service Accounting is installed and the default Cancellation status under Accounting, Configuration is set to Pending Cancellation. On cancellation, a request reaches first reaches Pending Cancellation status and then, after an invoice run, reaches Cancelled status.

4	Cancelled	On cancellation, a request reaches this status, unless CA Service Accounting is installed and the default Cancellation status under Accounting, Configuration is set to Pending Cancellation. **Note:** For information about the default Cancellation status and other CA Service Accounting-related settings that you define as part of the Accounting Configuration options, see the *Implementation Guide*.
6	Pending Resource Assignment	Applies *only* when CA Business Service Insight is integrated and a request that includes a service option of type application or agreement reaches Fulfilled status. Such a service option requires provisioning; therefore, the request is moved to Pending Resource Assignment status. **Note:** For more information about integrating with CA Business Service Insight, see the *Integration Guide*.
7	Resource Assigned	Applies *only* when a request is at Pending Resource Assignment status (the previous status in this table). Once the service option in such a request is provisioned, the request moves to Resource Assigned status.
100	Not Submitted	User-initiated status. When creating a request, users can optionally click Save as Request to save the request in Not Submitted status, rather than submitting the request.
101	Not Submitted – Cart	Catalog system-assigned status. When a request item is added to cart, it reaches this status. **Note:** This status appears only in reports, not on the GUI.
103	Not Submitted – Rejected	System-assigned status. After a service in a request is rejected, the request eventually reaches a Not Submitted – Rejected status. The request is returned to the requestor's open queue for review, modification, and re-submission.
104	Not Submitted – Approved	Catalog system-assigned status. When the same request contains both approved and rejected services, the approved services eventually reach a Not Submitted - Approved status. This status helps the user to differentiate between a previously rejected request items and approved request items. In this case, the entire request is in Not Submitted – Rejected status.

200	Submitted	User-initiated status. A request reaches this status when a requestor submits the request. There are three possibilities, depending on the type of approval process configured for each service in the request, as follows:

- For the Workflow driven approval process, a rule action is triggered to activate approval process definition, which determines the approver for the service and changes the status to Pending Approval.
  By default, all Catalog Content services use the CA Workflow driven approval process.

- For the System Approval process, the next action and status are determined by the Catalog system, based on the approval level assigned to the service and the management hierarchy of the user for whom the service is requested.

- For the No approval process, if the service is configured for fulfillment, then the status eventually changes to Pending Fulfillment. If the service is not configured for fulfillment, then the status eventually changes to Fulfilled. When all services in the request reach the Fulfilled status, the request reaches the Completed status.

201	Re-submitted	User-initiated status. When a rejected request is submitted, it reaches a Re-submitted status. The eventual flow is similar to Submitted status.

400	Pending Approval	Catalog system-assigned status. A request reaches this status when one or more services in a request require approval. There are two possibilities, depending on the type of approval process configured for each service in the request, as follows:

- For the Workflow driven approval process, a rule action is triggered to activate approval process definition, which determines the approver for the service and changes the status to Pending Approval.

- By default, all Catalog Content services use the Workflow driven approval process.

- For the System Approval process, the next action and status are determined by the Catalog system, based on the approval level assigned to the service and the management hierarchy of the user for whom the service is requested.

600	Rejected	User-assigned status. If a service in the request is configured for approval, a pending action is assigned to the approver. The approver can optionally either approve or reject each service in the request. When a service is rejected, the entire request is rejected, and the request is returned to the requestor's queue for review, modification, and re-submission.

800	Approved	User-assigned status. If a service in the request is configured for approval, a pending action is assigned to the approver. The approver can either approve or reject each service in the request.
		When a service is approved, it can either reach Approval Done status or get re-assigned to Pending Approval status if further approval is required. When all services in the request have received all required approvals, the request reaches the Approval Done status.
801	Approval Not Needed	Catalog system-assigned status. If a service in the request is configured for no approval, the service reaches this status after the request is submitted. The service eventually reaches Approval Done status.
999	Approval Done	Catalog system-assigned or CA Workflow-assigned status. When a service in a request is approved by all required approvers, that service reaches Approval Done status. When all services in a request reach Approval Done status, the entire request is completely approved and eventually reaches Approval Done status.
		The next status after Approval Done is either Pending Fulfillment or Fulfilled.
		Using Pending Fulfillment allows for additional fulfillment business processes to be followed by using CA Workflow process definitions.
		If the request does not require fulfillment, then it reaches a state of Fulfilled.

1000	Pending Fulfillment	Catalog system-assigned or CA Workflow-assigned status. In a request containing multiple services, the services may belong to any or all of the following categories:

■ If a service in the request is configured for fulfillment, a fulfillment process definition is triggered that determines the fulfiller for the service option and changes its status to Pending Fulfillment. Using Pending Fulfillment allows for additional fulfillment business processes to be followed.

■ If the request does not require fulfillment, then it reaches a state of Fulfilled.

Once the service is configured for pending fulfillment, one of the following actions occurs:

■ If a service option in the request is configured for simple fulfillment, a fulfillment process definition is triggered, and the new status is set to Pending Fulfillment.

■ If a service in the request is configured for complex fulfillment, a fulfillment process definition is triggered, and the new status is set to Check Availability (the next status in this table).

1001	Check Availability	CA Workflow-assigned status. This status applies when the Catalog system is configured for complex fulfillment.

When a service option in the request is configured for fulfillment and belongs to a hardware or software category, an associated rule action triggers a fulfillment process definition, which typically assigns a pending action to an IT Services User (fulfiller). The fulfiller can choose between Filled from Inventory and Not Filled from Inventory, or can bypass fulfillment completely by selecting Fulfillment Cancelled or Fulfilled.

1002	Filled From Inventory	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment (described earlier in this table).
		When an IT Services user is assigned a service option that is in Check Availability status, the fulfiller can choose between Filled from Inventory and Not Filled from Inventory.
		This status is set in one of the following ways:

- If CA APM is integrated with CA Service Catalog, this status is set when an asset is assigned by a CA APM user.

- If CA APM is not integrated with CA Service Catalog, the IT Services user should manually set the status to Filled from Inventory after checking to see if the requested items are available.

Note: For more information about integration with CA APM, see the *Integration Guide*.

1003	Not Filled From Inventory	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment (described earlier in this table).
		When an IT Services user is assigned a service option that is in Check Availability status, the fulfiller can choose between Filled from Inventory and Not Filled from Inventory.
		This status is set in one of the following ways:

- If CA APM is integrated with CA Service Catalog, this status is set when an asset is not available in CA APM.

- If CA APM is not integrated with CA Service Catalog, the IT Services user manuallys set the status to Not Filled from Inventory when one or more requested items are not available.

When the status of the service option is set to Not Filled From Inventory, the status eventually reaches Pending Procurement, meaning that the requested hardware or software must be ordered.

1004	Ordered	User-assigned status. This status applies when Catalog system is configured for complex fulfillment. Once the status of a service option is set to Pending Procurement, the IT Services user sets its status to the *actual* status of the ordered item, such as Ordered.
1005	Backordered	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Pending Procurement, the IT Services user sets its status to the *actual* status of the ordered item, such as Backordered.

1006	Shipped	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Pending Procurement, the IT Services user sets its status to the *actual* status of the ordered item, such as Shipped.
1007	Received	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Pending Procurement, the IT Services user sets its status to the *actual* status of the ordered item, such as Received.
		When the status of the service option is set to Received, it eventually reaches a status of Check Availability for re-assignment by an IT Services user or by a CA APM user (if CA APM is integrated with CA Service Catalog).
1008	Order Cancelled	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Pending Procurement, the IT Services user sets its status to the *actual* status of the ordered item, such as Order Cancelled.
		When the status of the service option is set to Order Cancelled, it eventually reaches a status of Check Availability for re-assignment by an IT Services or user or by a CA APM user (if CA APM is integrated with CA Service Catalog).
1012	Pending Procurement	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment.
		When the status of the service option is set to Not Filled From Inventory, it eventually reaches a status of Pending Procurement, meaning that the requested hardware or software must be ordered.
1013	USD Change Order Opened	CA Workflow-assigned status. This status applies when the Catalog system is configured for complex fulfillment and a rule action for creating a CA Service Desk Manager change order is enabled.
		When an IT Services user is assigned a service option in Check Availability status, and that user sets the status of the service option to Filled from Inventory, and if CA Service Desk Manager is integrated with CA Service Catalog, the status is set to USD Change Order Opened and a Change Order is created in CA Service Desk Manager.
		Note: For more information about integration with CA Service Desk Manager, see the *Integration Guide*.

1015	Notified IT Services	CA Workflow-assigned status. This status applies when the Catalog system is configured for complex fulfillment.
		When an IT Services user is assigned a service option in Check Availability status, and that user sets the status of the service option to Filled from Inventory, if CA Service Desk Manager is not integrated with CA Service Catalog, a process definition is triggered that assigns a pending action to an IT Services User and sets the status to Notified IT Services.
1016	Being Staged	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Notified IT Services, the IT Services user sets its status to the *actual* status of the requested service option, such as Being Staged.
1017	Staged	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Notified IT Services, the IT Services user sets its status to the *actual* status of the requested service option, such as Staged.
1018	Being Configured	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Notified IT Services, the IT Services user sets its status to the *actual* status of the requested service option, such as Being Configured.
1019	Configured	User-assigned status. This status applies when the Catalog system is configured for complex fulfillment. Once the status of a service option is set to Notified IT Services, the IT Services user sets its status to the *actual* status of the requested service option, such as Configured.
1020	USD Request Opened	CA Workflow-assigned status. This status applies when the Catalog system is configured for simple fulfillment and a rule action for creating a CA Service Desk Manager change order is enabled.
		When an IT Services user is assigned a service option in Pending Fulfillment status, and if CA Service Desk Manager is integrated with CA Service Catalog, the status is set to USD Request Opened and a request is created in CA Service Desk Manager.
		Note: For more information about integration with CA Service Desk Manager, see the *Integration Guide*.

1999	Fulfillment Cancelled	User-assigned or Catalog system-assigned status. This status applies when one of the following is true:

■ The IT Services user sets the status to Fulfillment Cancelled when the service option does not need to be fulfilled.

■ The Catalog system sets the status to Fulfillment Cancelled when a user cancels a request that has been approved but has not been fulfilled.

When all services are set to Fulfillment Cancelled, the request eventually reaches Cancelled status.

However, when the request contains services that are in both Fulfilled and Fulfillment Cancelled status, the request eventually reaches Completed status.

2000	Fulfilled	User-assigned status. This status applies when one of the following is true:

■ The IT Services user sets the status of the service option to Fulfilled.

■ If CA Service Desk Manager is integrated, the process definition in CA Service Desk Manager sets the status to Fulfilled when an associated change order is closed in CA Service Desk Manager.

When all services are set to Fulfilled, the request eventually reaches Completed status.

When the request contains services that are in both Fulfilled and Fulfillment Cancelled status, the request eventually reaches Completed status.

3000	Hold	User-assigned status. This status applies when either an administrator or the Catalog system sets the status of the service option to Held, as explained in How to Hold and Resume Requests, Services, or Service Options (see page 598). When an item is held, no status changes can occur for it until you *resume* it; that is, change its status to Resume.

Note: When a service option is held, the monitoring of time is stopped for any request SLAs (see page 254) attached to the service option, to prevent related SLA warnings and SLA violations from being issued inordinately.

4000	Resume	User-assigned status. This status applies when an administrator sets the status of a previously held service option to Resume, as explained in How to Hold and Resume Requests, Services, or Service Options (see page 598). When resumed, service options that were formerly held move to a temporary Resume status before automatically moving back to the status they were at before they put on Hold.

Note: When you resume a previously held service or service option, the monitoring of time is resumed for any request SLAs (see page 254) attached to the service or service option.

Request Life Cycle

After you have submitted your cart, you have launched your new request into its life cycle. Each requested service option in the request has a status that reflects its current step in the life cycle of the request. The overall request also has a milestone status that reflects the aggregate values of the statuses of the requested service options. The milestone statuses are:

- Not Submitted
- Submitted
- Pending Approval
- Approved/Rejected
- Pending Fulfillment
- Fulfilled
- Completed/Cancelled

Each status range contains various detailed status values. The status values are customizable by your catalog administrator so the values given as examples may not match what you view in your system. During the approval and fulfillment phases for your request, you might receive emails informing you of the progress of your request through its life cycle. The business process for approval and fulfillment, the status values used and whether you receive emails depends on your service provider's approval and fulfillment business processes.

As an example, suppose that you request a Desktop service which includes a Standard Desktop service option and Microsoft Visio service option. Your request includes one service which contains two service options. If your service provider requires manager approval, the first step is for your manager to approve or reject your request for the Desktop service (approval is at the service level). After approval phase is complete and the two service options (Standard Desktop and Microsoft Visio) begin the fulfillment phase. If the fulfillment process for these two service options includes checking available inventory, an asset manager might be notified to check inventory.

Assuming both the Standard Desktop and a Microsoft Visio license are found in inventory, the status of each service option changes to Filled From Inventory. The next step in the fulfillment process might be to notify IT Services to configure and deliver the Standard Desktop and Microsoft Visio so the status of each service option changes to Notified IT Services. After each service option is configured and delivered to you the status of each service option changes to Fulfilled. After each service option in the request is set to Fulfilled, the status of all service options is set to Completed and the request is considered completed or closed.

For this example, for each requested service option, you can view the following status values over time (in chronological order):

- Not Submitted - Cart

- Submitted

- Pending Approval

- Approved

- Approval Done

- Pending Fulfillment

- Check Availability

- Filled From Inventory

- Notified IT Services

- Fulfilled

- Completed

Request List Pages

Request list pages display all requests or a subset of filtered requests, such as open, pending, and completed requests. A request list page appears when users click Home, Requests or when users select any of the following options from the Requests page:

- Open Requests

- My Recent Requests

- Completed Requests

- Pending My Action

- Advanced Search–after you search for requests (see page 587) and view the results

When a user selects one of these options, the Catalog system opens a request list page that displays the requests in a table, one request per row. Each column in the table displays an attribute of the request, such as the request ID, name, priority, or status. Administrators with the Service Provider role (spadmin) can optionally configure the display of columns on request list pages (see page 513).

View Recent Requests

The My Recent Requests section of the Requests window displays information about the requests either made for you or by you.

Administrators can configure the display of columns (see page 513) on the My Recent Requests page to display some or all of the following columns:

- ID - Request ID assigned by the system.

- Name - Request name chosen by you.

- Date Created - Date and time the cart or request was originally started.

- Date Modified - Date and time the request was last modified.

- Status - Current status of the request.

Using the My Recent Requests list you can:

- View a request's status history by clicking the Status column value

- View the details of a request

View Open Requests

The Requests Lookup section of the Requests window displays the number of open requests either requested for you or requested by you. An open request is one whose status is not Completed or Cancelled. To view the list of Open Requests, click the Open Requests link. The Open Requests window appears. The functionality available through the Open Requests window is also available on other windows and is described in other sections of this chapter.

Administrators can configure the display of columns (see page 513) on the Open Requests page to display some or all of the following columns:

■   ID - Request ID assigned by the system.

■   Name - Request name chosen by you.

■   Requested By - The user who created this request.

■   Requested For - The user or account for which this request was created.

■   Priority - The priority set for the request.

■   Date Created - Date and time the cart or request was originally started.

■   Date Modified - Date and time the request was last modified.

■   Status - Current status of the request.

Using the Open Requests list you can:

■   View a request's status history by clicking the Status column value

■   View the details of a request

■   Edit the request

■   Send an email about the request

The following functionality is available if your role permits:

■   Delete or cancel a request

■   Ignore, retry, or override an alert

View Completed Requests

The Requests Lookup section of the Requests window displays the number of completed requests either requested for you or requested by you. A completed request is one whose status is Completed or Cancelled. To view the list of Completed Requests, click the Completed Requests link. The Completed Requests window appears. The functionality available through the Completed Requests window is also available on other windows and is described in other sections of this chapter.

Administrators can configure the display of columns (see page 513) on the Completed Requests page to display some or all of the following columns:

- ID - Request ID assigned by the system.

- Name - Request name chosen by you.

- Requested By - The user who created this request.

- Requested For - The user or account for which this request was created.

- Date Created - Date and time the cart or request was originally started.

- Date Completed - The date the status of all the service options in the request was set to Fulfilled or Fulfillment Cancelled.

- Status - Current status of the request.

Using the My Recent Requests list you can:

- View a request's status history by clicking the Status column value.

- View the details of a request.

- Send an email about the request.

The following functionality is available if your role permits:

- Cancel all the services in the request.

Search for Requests

You can use the Request Lookup section of the Requests window to search for requests by request ID or according to the predefined categories on the screen. Furthermore, you can click Advanced Search (see page 588) to search for requests by additional (advanced) criteria. You can view only the requests permitted by your role.

To search for requests

1. Click Home, Requests.

   The Request Lookup and My Recent Requests boxes appear.

2. Check for the request of interest in the My Recent Requests list.

   ■ If you see the request, click it to view its details and optionally edit. You can skip the remaining steps.

   ■ If you do not see the request, go to the next step

3. If you know the ID of the request you are searching for, enter that ID in the Find Request By ID field, click Go, and do the following:

   ■ If the search results return the request, click it to view its details and optionally edit.

   ■ If the Catalog system cannot find the request ID, an error message appears.

   Verify the request ID and try again, or go to the next step.

4. Click the link for the category of requests that you want to view; for example, Open Requests, Completed Requests, Pending My Action, and so forth.

   When you click the link, the requests in that category appear. The tool bar above them includes any applicable actions that you can perform on requests pending action (see page 604).

5. Click View All Requested Items to view a list of all service options that you have ordered in your requests pending action.

You have searched and viewed requests.

Advanced Search for Requests

You can create and run advanced search queries to help determine the requests that need special attention or meet certain criteria. Running advanced search queries helps verify that you know about and act on all such requests, especially when the total number of requests in your system is large.

Note: This note applies only to upgrades, not new installations. If you are an administrator who saved queries from the previous release, verify that your saved requests continue to work in this release. You may need to update or replace your saved queries from previous releases.

To perform an advanced search for requests

1. Click Home, Requests.

   The Requests page appears.

2. Click Advanced Search in the Request Lookup box.

   The Request Lookup box changes to the Advanced Search box, and other elements of the Requests page also change accordingly.

3. Do one of the following:

   - Click the Search icon (by default, a magnifying glass) to enter a new advanced search query, as explained in the next step.

   - Click the Save icon (by default, a folder with down arrow) to save an advanced query that you have entered and, if applicable, share it, as explained in a later step.

   - Click the Load icon (by default, an open folder with papers) to view the list of saved advanced search queries, as explained in a later step.

   Go to the step that matches your action–that matches the icon you clicked.

4. Click the Search icon (by default, a magnifying glass) in the Advanced Search box to enter a new advanced search query, if applicable; otherwise, skip this step.

   The fields next to the Search icon expand so that you can enter an advanced search condition in the format *property - operator - value*, as follows:

   a. In the first (property) search field (a drop-down list), select the property for you want to search for, such as Action By User ID, Business unit Name, Request Priority, Service Option Element Name, and so forth.

   b. In the second (operator) search field (a drop-down list), select the operator to use for your search, such as Equals, Not Equals, Contains, Starts With, Ends With, In, and so forth.

      The operators that appear change as needed to match the property that you selected.

      In is a special operator that you can use to select or specify *multiple* values as search criteria, as explained later in this procedure.

   c. In the third (value) search field, type or select your search criteria, as follows:

      ■ For most properties, you type custom search criteria directly into this field. Examples include actual request IDs, user names, and business units.

         If you select any operator except *In* in the operator search field, specify a *single* value in the value field, to search for a single value.

         If you select *In* in the operator search field, you can specify multiple values in the value field, to search for multiple values.

         For example, to search for the request whose ID is 101, specify the following in the search fields:

         property: Request ID   operator: Equals   value: 101

         Similarly, to search for three requests whose IDs are 101, 102, and 103, specify the following in the search fields:

         property: Request ID   operator: In   value: 101,102,103

         Note: Use a comma (,) to separate multiple values in the value field.

■ Consider doing the following if you are a Service Delivery administrator and if you plan to share this advanced search query with other users:

For properties that include the user ID, specify the value of this field as the variable named $USERID$.

For properties that include the business unit ID, specify the value of this field as the variable named BU.

If you do so, when other users run this query, the Catalog system dynamically replaces the variable with the user ID or business unit ID of the user running the query, rather than your own user ID or business unit. This dynamic replacement is especially useful for complex advanced search queries that other users may find beneficial but may lack the expertise to create. For similar examples, see the predefined advanced search queries that appear when you click the Load icon in the Advanced Search box.

■ For certain properties, this field changes to a drop-down list, and you select the value you want from the list. For example, if you are searching for a property related to the request status, you select the request status you want from the drop-down list. If your implementation includes custom statuses, they appear in this list.

If you select any operator except *In* in the operator search field, select a *single* value from the list, to search for a single value.

If you select *In* in the operator search field, you can select multiple values from the list, to search for multiple values. To select additional values after you select the first value, press Ctrl and select the additional values.

For example, to search for requests whose status is Approved, specify the following in the search fields:

property: Request Current Status   operator: Equals   value: Approved

Similarly, to search for requests whose statuses are Approval Done or Approval Not Needed, specify the following in these fields:

property: Request Current Status   operator: In   value: Approval Done and Approval Not Needed

■ If you are searching for a property related to the dates, a calendar appears next to the field, and you select the date you want on the calendar.

If required, you can include multiple dates in your query by using operators such as Before or After, and by using multiple search conditions. For example, to find request created between July 27, 2011 and July 30, 2011, inclusive, specify both of the following conditions in your query:

property: Request Date Created   operator: After or On   value: 07/27/2011

property: Request Date Created   operator: Before or On   value: 07/30/2011

You have completed this search condition.

       d. Click the Plus sign (+) after the third search field to specify another search condition, if applicable. You can optionally specify multiple search conditions in an advanced search query. If you do not want to specify another search condition, go to the next step.

       e. Click the Remove (-) icon after the third search field of any additional search conditions that you want to remove from the advanced search query, if applicable.

You have finished specifying this advanced search query.

5. Click the Save icon (by default, a folder with down arrow) in the Advanced Search box to save an advanced query that you have entered and, if applicable, share it.

Perform this step if applicable; otherwise, skip this step.

The Save Query dialog appears.

Do the following:

       a. Enter meaningful data in the Name and Description fields.

All users can save advanced search queries for personal use.

       b. Click Share to share the query with other users, if applicable.

Only Service Delivery administrators can share queries. When a query is shared, it appears as a saved query for all users when they click the Load icon. All users in all business units can view and run shared queries, but cannot modify them.

Users with any role other than Service Delivery administrator can save their own advanced search queries but cannot share them. Such users can load and run either their own saved queries or the queries that the Service Delivery administrator has shared.

You have saved this advanced search query and, if applicable, have shared it.

The Save Query dialog closes, and you return to Advanced Search box.

6. Click the Load icon (by default, an open folder with papers) in the Advanced Search box to view the list of saved advanced search queries, if applicable; otherwise, skip this step.

    The Load Query dialog appears.

    Do the following:

    a. Filter the list to display all queries, shared queries only, or personal queries, by clicking the option you want in the View drop-down list.

    b. Click the query you want to select it and insert it into the Advanced Search box.

        If the query includes multiple conditions, you can optionally remove any that you do not want to include.

        The Catalog system includes several predefined queries at installation time. Service Delivery administrators can optionally unshare or delete any predefined queries not applicable to their implementation.

        Note: You cannot modify a saved advanced search query. However, you can optionally note the property, operator, and value specifications in the query and use them as model to create a new query.

    You have loaded the advanced search query.

    The Load Query dialog closes, and you return to Advanced Search box.

7. Click Search in the Advanced Search box to run the advanced search.

    The Catalog system returns the search results.

Note: Administrators can define the scope of user searches by clicking Administration, Configuration, User Default and specifying the User Search Scope as either the business unit or (Business Units) of the user or the entire Catalog system (Enterprise).

View and Modify Requests

You can view and modify the details of a request by clicking the request name from one of the lists of requests. The Request Details window appears.

The Request Details window displays similar information that was shown in the cart. See other sections of this chapter for details on functionality that is not described in this section. Depending on the current status of the request and your role, you can have different functionality available with this window.

The following functionality is available regardless of the status of the request:

- View the requested services and service options (by clicking Show Details)

- View several sections of additional information about the request

- Display and add Notes

- Display and add Attachments

- View the status history of the request or requested service options

- Refresh the display

- Copy the request as an unsubmitted request

- Email a copy of the request

- Edit the request's additional information sections

The following functionality is available if the request is in the *not submitted* phase:

- Submit the request for approval and fulfillment

- Delete the request

The following functionality is available if you have an assigned approval pending action for the request:

- Open the approval window

The following functionality is available if you have an assigned fulfillment pending action for the request:

- Open the fulfillment window

Edit the Request

To edit information about the request, click the Edit button. The Edit Request window appears.

The Edit Request window displays similar information that was shown in the cart. See other sections of this chapter for details on functionality that is not described in this section. Depending on the current status of the request and your role, you can have different functionality available with this window.

This window allows you to do these things:

- View the requested services and service options (by clicking Show Details)

- Alter several sections of additional information about the request

- Display and add Notes

- Display and add Attachments

- View assigned assets (if the service option is eligible for asset assignment and the request is in the fulfillment phase)

- Save your changes

The following functionality is available if the request is in the *not submitted* phase:

- Add new services from the catalog

- Delete a requested service from the request

The following functionality is available if the request has a status of Completed:

- Cancel a requested service

- Cancel the request which cancels all the services in the request

Add Services to a Request

If a request is in the *not submitted* phase, you can add additional services to the request. To browse the catalog and add services to the request, click the Add from Catalog button. You can browse and select services from the catalog the same way you did when originally adding services to your cart. When a service is selected and the Add to Request button is clicked, you are returned to the Edit Request window with the selected service added to the request.

View Assigned Assets

If CA APM is installed and a service option is eligible to have an asset assigned to it, the Actions column can contain an Assigned Assets gold brick icon. Clicking the Assigned Assets icon displays the CA APM Assign Assets window with the Assigned Assets Action selected and the list of assigned assets shown.

Note: To use the Assign Assets window, your user ID must be assigned to a role in CA APM with the correct security privileges.

Cancel a Service

If a service has a status of Completed and your role permits, the Actions column contains a Cancel link. Clicking the Cancel link changes the status of the service and all of its service options to Cancelled unless CA Service Accounting is also installed. If CA Service Accounting is installed, the status of the service and all of its service options changes to the value specified by your catalog administrator (Pending Cancellation or Cancelled).

Copy a Request

You can copy the request whose details you are viewing to use it as a template for a new request.

To copy a request

1. Click Home, Requests.

2. Click the list of requests that contains the request that you want to copy. For example, if you want to copy a completed request, click Completed Requests.

    The list of requests appears.

3. Click the name of the request that you want to copy.

    The Request Details window for this request appears.

4. Click the Copy button.

    You are prompted to enter the name of the new request.

5. Enter the new name and click OK.

A new request is created, with the same values in the request information fields and the same requested services as the request being copied.

In the request information fields, the Requested By user is the user who copied the request, and the Requested For user or account is copied from the original request.

The Edit Request window appears for the new request.

Note: The notes and attachments for the original request are not copied to the new request. The status of the new request is Not Submitted, so that you can delete the existing services or add new services in the request.

6. Complete the fields as needed and click Save.

Email the Request

To email the details of a request, click the Email button or link. The Email Request window appears. Enter the following information and click Send to send the email to the specified recipients.

■ Include Request Snapshot - If yes, details of the request are included in the email along with the Message Body. If no, then only the specified Message Body is sent.

■ To - You can specify one or more primary recipients for the email. You can use specific email addresses, user IDs, account names or user last name, first name values. If specifying more than one value, separate the entries with a semi-colon (;). You can also use the Click Here to Choose Account magnifying glass icon to display the Search window where you can search for users and accounts. When specifying an account name you are really sending the email to the email address on the account profile.

■ CC - You can specify one or more CC recipients for the email. Use the same syntax as the To field. To automatically send an email to either the user or account used in the Requested For or Requested By field for the request, select the appropriate check box.

■ BCC - You can specify one or more BCC recipients for the email. Use the same syntax as for the To field.

■ Subject - This text is used in the subject of the email.

■ Message Body - This text is used in the message body of the email. If you select the Rich Text check box a tool bar allowing extra rich text capabilities appears.

How Hold and Resume Actions Work

In CA Service Catalog, you can optionally *hold* a request, hold one or more services in a request, or hold one or more service options in a service. Holding requests, services, or service options may be required to make clear to anyone monitoring a request that the processing of the request, service, or service option has been *intentionally* suspended for a time. Reasons for holding include the following and others:

- Temporary unavailability of the requestor or approver

- Temporary unavailability of requested items, such as the items being in back-ordered status

When a service option is held, the monitoring of time is stopped for any request SLAs (see page 254) attached to the service option, to prevent related SLA warnings and SLA violations from being issued inordinately.

You hold a request, service, or service option by changing its status to Hold. When an item is held, no status changes can occur for it until you *resume* it; that is, change its status to Resume. When you resume an item that was held, it returns to its last previous status before it was held.

If you hold only selected services within a request or only selected service options within a service, the services or services options that you did not hold continue the normal request life cycle, except that they cannot reach Completed status until the held items are resumed and are also completed.

When resumed, service options that were formerly held move to a temporary Resume status before automatically moving back to the status they were at before they put on Hold. Checking Resume status is useful for history purposes and to help you set up optional rules and actions, automated processing using CA Process Automation, or automatic email notifications.

Only CA Service Catalog administrators and other users with the required access control setting can hold and resume requests. The following note explains the required access control setting.

Note: The access control option named Access Control: Hold/Resume Request under Service Builder, Configuration determines which roles have the ability to hold and resume requests pending action assigned to other users. For details, see the Service Builder configuration settings in the *Implementation Guide*.

Hold and Resume Requests, Services, or Service Options

In CA Service Catalog, you can optionally hold and resume (see page 597) requests, services, or service options. You can *hold* a request, hold one or more services in a request, or hold one or more service options in a service. Holding requests, services, or service options may be required to make clear to anyone monitoring a request that the processing of the request, service, or service option has been *intentionally* suspended for a time.

Note: The access control option named Access Control: Hold/Resume Request under Service Builder, Configuration determines which roles have the ability to hold and resume requests pending action assigned to other users. For details, see the Service Builder configuration settings in the *Implementation Guide*.

To hold and resume requests, services, or service options

1. Click Home, Requests.

2. Click Pending My Action or Open Requests. Alternatively, click Advanced Search for requests.

   The Requests Pending Action window appears, or the Open Requests window appears.

   Click the Hold/Resume icon for the request you want.

   The Hold/Resume window appears, showing the details for the selected request, including a Requested Services section at the top of the screen. That section lists all services and service options in the request. If the request contains multiple services or service options, you can hold or resume them either individually or in groups of two or more.

3. To *hold* one or more services or service options, select the services or service options and click Hold.

   Each service option you selected is marked *to be held* by your user ID.

   If a request is already held, the status indicating "held by *username*" appears next to the Item Status field of the request item, and therefore you cannot hold it again. The *username* identifies the user who has held the request.

   Note: When you hold a service or service option, the monitoring of time is paused for any request SLA (see page 254) attached to the service or service option.

4. To *hold* an entire request, select the All check box for the request and click Hold.

   All service options in the request are marked *to be held* by your user ID.

5. Add a note to the request for the services or service options that you held explaining why you held them. Provide any details or conditions that help other administrators determine when to resume a service or service option. For example, you may add a note such as the following: "The requested laptop is back-ordered. Resume this service option when we receive the laptop and deliver it to the user."

6. Before resuming a service or service option, review the request details explaining why it was held, including the notes and the Request Status History. Most importantly, review whether the service or service option was held by the Catalog system or by a specific user ID, and review the reasons for the hold. Verify that all reasons have been addressed before resuming the request.

7. To *resume* one or more services or service options, select the services or service options and click Resume.

8. **Note:** When you resume a previously held service or service option, the monitoring of time is resumed for any request SLA attached to the service or service option.

    The only services that you can resume are those already marked as held.

9. Click Save to perform the hold or resume action.

    Your action is confirmed: Any service option that you have held is marked as held by your user ID. Similarly, any service option that you have resumed is no longer marked as held, meaning that it continues the normal request life cycle.

Optionally, you can do one of the following:

■   Perform another action on the Hold/Resume window

■   Access another window

■   Set up automatic email notifications for held and resumed requests (see page 600)

Automatic Email Notifications for Held and Resumed Requests and Service Options

You can optionally hold and resume requests, services, or service options (see page 598). In addition, you can optionally configure and use the events, rules, and actions for sending automatic email notifications when a request, service, or service option is held or resumed.

The events named Request Change and Request/Subscription Item Change each include two rules related to hold-resume functions: When Status is Hold and When Status is Resumed.

For the Request Change event, the When Status is Hold rule includes an action that sends an automatic email notification when a request is held. Similarly, the When Status is Resumed rule includes an action that sends an automatic email notification when a request is resumed.

For the Request/Subscription Item Change event, the When Status is Hold rule includes an action that sends an automatic email notification when a service option is held. Similarly, the When Status is Resumed rule sends includes an action that sends an automatic email notification when a service option is resumed.

You can optionally enable and configure these events, rules, and actions (see page 26) and others to meet your requirements.

Cancel a Request

You may need to cancel a request for any of several possible reasons, for example, if the wrong items were requested or if the requested items are no longer wanted or needed.

Only CA Service Catalog administrators and other users with the required access control setting can view the Cancel button and use it to cancel requests. The access control option named Access Control: Cancel Request under Service Builder, Configuration determines which roles can cancel requests. For details, see the Service Builder configuration settings in the *Implementation Guide*.

In addition, you can cancel a request *only* if the status of the request has not exceeded the status specified by the Allow Cancellation Through setting in the Service Builder, Configuration. This setting defines the status through which a request can be canceled. After the request moves to the next status, it cannot be canceled.

Note: By default, for the event named Request/Subscription Item Change, for the rule named When Status is Canceled, an action is enabled that terminates CA Workflow instances and aborts CA Process Automation instances when a request is canceled. For best product performance, keep this default setting. For details about this rule and its action, see the *Integration Guide*.

To cancel the request

1. Click Home, Requests.

2. Click the list of requests that contains the request that you want to cancel. For example, if you want to cancel a request pending action, click Pending My Action.

   The list of requests appears.

3. Click the request that you want to cancel.

   The Request Details window for this request appears.

4. Click the Cancel button.

   Note: If the requirements for cancellation mentioned earlier are not met, the Cancel button is disabled, and the request cannot be canceled. Moreover, items in the same request may be at different statuses: some items may have exceeded the Allow Cancellation Through status while others have not yet done so. In such cases, you can cancel *only* the items that have *not* exceeded the Allow Cancellation Through status.

5. Confirm the cancellation and click OK.

View Request Tracking

Depending on your role and the configuration options set by your catalog administrator, when viewing Request Details, you can also view a Tracking tab. Click the Tracking tab to display the Process Instance List. The Process Instance List displays information regarding any CA Workflow or CA IT PAM process instances related to the requested services and service options.

The Process Instance List displays these columns:

- Request Item/Service - This column indicates the name of the service option or service that has an associated CA Workflow or CA IT PAM process instance.

- Process Name - The name of the CA Workflow process definition or CA IT PAM process on which this process instance is based.

- Status - The status of the process instance.

- Start Time - The time the process instance was started.

- End Time - The time the process instance ended.

  Note: This column can be populated only when the process instance status is Completed or Terminated.

- Duration - The time in days, hours, minutes and seconds that the process instance has been running.

Use this window to view the list of processes generated by the workflow engine for the request.

View Process Instance History

To view the history of a process instance, click the process instance name. The Process Instance History window appears. This window displays detailed information about each step in the CA Workflow process instance. By default, only the workitems are shown. Workitems are CA Workflow steps (nodes) that use a system or person to accomplish a task. You can tell when and to whom each workitem is assigned and completed.

By clicking the Show All link, you can view even more details about the CA Workflow process instance history.

View a Request Audit Trail

Depending on your role and the configuration options set by your catalog administrator, when viewing Request Details, you can also view an Audit Trail tab. Click the Audit Trail tab to display the Audit Trail list. The Audit Trail list displays each service option element and its status changes over time. Each service option is made up of service option elements. In addition, you can view the request status changes listed.

To view the detailed old and new values of each field related to the request or service option element, click the Detail icon in the Details column. The Change Event Details window appears. You can display only the transactions where the field actually changed value or display all transactions.

Work With Assigned Assets

If your CA Service Catalog implementation integrates with CA APM, you can work with assigned assets, to perform the following tasks:

- Assign an available asset to a requested service option

- Mark a requested service option as not filled from inventory

- Remove an asset assignment

For details, see the *Integration Guide*.

View All Requested Items

To view all service options that have been requested and submitted for approval and fulfillment, click the View All Requested Items link in the Request Lookup section of the Requests window. If CA Service Accounting is installed, this list canals include the service options to which the user has been subscribed. The display displays then number of times each service option has been ordered, where *ordered* means the service option has been subscribed or a request for the service option has been submitted. If a service option is contained in a request that has not been submitted or if the service option is still in the cart it cannot be included in the count.

To view the requests or subscriptions for the service option, click the link indicating how many times you have ordered the service option. The list of requests containing the service option appears.

To view the subscriptions for service option for the user's account, click the View Subscriptions button.

Note: If the user has not subscribed to the service option or if CA Service Accounting is not installed, the View Subscriptions button is not displayed.

Handle Requests Pending Action

When a cart or request is submitted, it enters the approval and subsequent fulfillment phase. Your service provider can determine the business processes for approval and fulfillment of the services and service options in the catalog. During approval and fulfillment, it is likely that a person is assigned an approval or fulfillment task.

If there is a request approval or fulfillment task assigned to you or to a group of which you are a member, you may receive an email notifying you that a task has been assigned to you. You can also view the requests which have an approval or fulfillment task assigned to you by viewing your Pending Action list. To view your Pending Action list, click the Pending My Action link in the Request Lookup section of the Requests window.

Note: This section of the Requests window also displays how many requests are waiting for you complete an approval or fulfillment task.

The Requests Pending Action list displays the list of requests which have an approval or fulfillment task assigned to you. You can optionally filter this list of assignments by your user name, your group name, or all.

The Requests Pending Action list displays these fields:

■   ID - Request ID assigned by the system

■   Name - Request name chosen by the user

■   Requested By - The user who created this request

■   Priority - The priority set for the request

■   Date Created/Modified - Date and time the request was created and was last modified

■   Group/Taken By - If a request pending action has been assigned *to* a group, the name of the group name appears in this column. If a user has *taken* the request pending action, the user name and group information are listed.

■   Delegation - If a request pending action has been delegated, this column displays the names of the user who delegated it and the user to whom it was delegated.

■   Status - Current status of the request

Using the Requests Pending Action list you can:

■   View the status history of a request (including its pending actions) by clicking the Status column value

■   View the details of a request

■   Open the approval or fulfillment window for a request

■   Edit the additional information sections for a request

■   Send an email about the request

- Take ownership of a request pending action assigned to a group to which you belong

- Return ownership of a request pending action to such a queue, after you have previously taken the request from the queue but have not approved or fulfilled the request

- Transfer or delegate a request pending action to another user in your organization

- Override, retry, or ignore requests that have become stuck

The ability to transfer or delegate requests pending action is especially helpful when the user originally assigned to approve or complete the request become unavailable due to emergency, illness, vacation, or another reason. Similarly, the ability to take and return requests pending action to a group queue (if one is used) helps maintain a balanced workload among team members. Together, these abilities help all users within your organization work together as a team to complete requests in a timely manner, even when challenges occur that require changes in plans.

Request administrators can *search* for all requests, including requests assigned to other users. Request administrators have one of the following roles: Service Delivery Administrator, Super Business Unit Administrator, Administrator, Catalog Administrator, and Request Manager. If the requests are already assigned to a user for approval or fulfillment, administrators can either complete these requests or transfer them to other users, including themselves.

For a complete list of rules for handling request pending action, see Rules for Requests Pending Action (see page 606).

During the approval or fulfillment process, requests may occasionally become *stuck*, typically due to system error or user error. Here, *stuck* means that the request has been approved for promotion to the next level, but nonetheless remains in its current status. When automatic retry is enabled, in rare cases, requests can remain stuck even after all retries have been attempted. To improve the efficiency of request management and reduce system load, it is recommended that you override, retry, or ignore stuck requests as soon as you become aware of them. For details, see Ignore, Retry, or Override Alerts (see page 627).

Rules for Requests Pending Action

In this topic, the term *request pending action* refers to one or more services or service options in a request that are pending approval or pending fulfillment. In both single-service and multi-service requests, all services are part of the request, but the term request pending action applies only to the services or service options that are pending approval or pending fulfillment. Specifically, pending *approval* status applies to *services* only, while pending *fulfillment* status applies to *service options* only.

Both CA Service Catalog administrators and non-administrator users can transfer or delegate requests pending action in their own queues to other users. When you either transfer or delegate a request pending action, the user who receives the request becomes the new assignee. The major difference between transferring and delegating follows: When you delegate a request, it stays in both your queue and the new assignee's queue. In contrast, when you transfer a request, it leaves your queue and moves to the new assignee's queue.

Thus, as an administrator, you may decide to delegate a request pending action if you want to monitor it within your own queue. Conversely, you may decide to transfer a request pending action if you do not need to monitor it so closely. Keep in mind, however, that after you delegate a request, the new assignee (User B) or an administrator may transfer the request to someone else (User C). In such cases, when User B or another transfers the request, the request moves from both your queue and User B's queue to User C's queue only.

Only requests pending action assigned to a user can be delegated. When a request pending action is assigned to a user and a group, only the one that is assigned to a user can be delegated. In other words, the one assigned to a group cannot be delegated.

A request pending action can be delegated only once. In other words, re-delegation is not permitted. This limitation includes auto-delegation; in other words, a request that has already been delegated manually cannot be auto-delegated. Therefore, before you delegate a task to a user, ensure that the user does not have auto-delegation enabled. However, the request can be transferred multiple times. You can delegate requests to an individual user only, not to a group.

A request pending action can be transferred or completed by the primary assignee of a pending action, by administrators, and by a user with the required access control settings; details about these settings are explained in the topics that follow. A request pending action can be transferred to only one user or one group.

Both administrators and non-administrative users can also take ownership of a request pending action assigned to a group queue to which they belong. Conversely, users can return to the group queue a request pending action that they have previously taken but have not completed.

A request pending action can only be returned by the primary owner of a taken pending action. The administrator (or any other user) cannot return a request pending action on another user's behalf.

Approve or Reject Your Own Requests Pending Action

Approving or rejecting requests pending action is an essential step in the process of managing requests from creation through fulfillment. All CA Service Catalog users can approve or reject   requests pending action assigned to them; that is, requests in their own Requests Pending Action queue. However, only administrators and users with the required access control setting can approve or reject other users' requests pending action (see page 610).

Note: The Approve Request window displays similar information to that shown in the Edit Request (see page 594) window.

To approve or reject your own request pending action

1. Click Home, Requests.

    Your recent requests are listed, along with links to perform other actions; for example, to view requests pending your action.

2. Click Pending My Action.

    The Requests Pending Action window appears, listing the requests in your queue to be approved, rejected, fulfilled, and so forth.

3. Locate the request and click the Approve/Reject icon in the Actions column.

    The Approve Request window appears.

4. View the details of requested services, including their statuses.

5. Focus on the services for which you are assigned an approval task.

    The status of each service option matches the status of the service during the approval phase of the request life cycle.

    Note: Typically, the Status drop-down list includes several options; the names of these options and the order in which they appear are specified in the requestshared.xml file. To update the values and the order of the options in the Status drop-down list, modify the requestshared.xml file to include the status values you want in the order that you want them to appear. For details about modifying the requestshared.xml file, see the "Customizing" chapter of the *Implementation Guide*.

6. Decide whether to approve or reject the request, and then do all of the following that apply:

    ■ As a best practice, if you do not want to approve one or more services, add a note explaining both why the service is being rejected. If applicable, also explain the changes required for the request to be approved.

    ■ Optionally add other notes or attachments as supporting documents.

7. In the Item Status drop-down list, select the appropriate status to approve or reject each service that you are concerned with, and click OK. The statuses are typically named Approved and Rejected, respectively.

■   Rejecting a service rejects the entire request. The request is returned to the user who submitted it. That user decides whether to update the request and resubmit it or accept the rejection as final.

■   Once all services in the request are approved, the request moves the request to the next phase of the request life cycle, for example, the second level of approval (if applicable) or the fulfillment stage. In other words, after you have approved a service, your service provider's approval processes may require a second level of approval in which case the second level approver is assigned an approval task.

If the approval task was assigned to more than one user or a group, once any assigned user completes that pending action, the request is removed from the Pending Action list for all assigned users. If the request contains multiple services, it is not removed until either all services are approved or one service is rejected.

You are returned to the Request Details window. If you return to the Pending Action list, you can see that the request is removed from your list.

Approve or Reject Other Users' Requests Pending Action

Approving or rejecting requests pending action is an essential step in the process of managing requests from creation through fulfillment. All CA Service Catalog users can approve or reject their own requests pending action (see page 608); that is, requests in their own Requests Pending Action queue. However, only CA Service Catalog administrators and other users with the required access control setting can approve or reject requests pending action assigned to other users. The following note explains the required access control setting.

Note: The access control option named Access Control: Proxy Action under Service Builder, Configuration determines which roles have the ability to approve, reject, fulfill, or transfer requests pending action assigned to other users. For details, see the Service Builder configuration settings in the *Implementation Guide*.

To approve or reject other users' pending action

1. Click Home, Requests.

    Your recent requests are listed, along with links to perform other actions; for example, to view requests pending your action.

2. In the Request Lookup section, click Advanced Search.

    The Advanced Search window appears.

3. In the Query drop-down list, do the following:

    a. Ensure that Action Requests is selected.

    b. Optionally, in the Userid field, enter the user ID of interest.

    c. Click Search.

    The Search Results window appears, displaying all requests pending action.

4. In the Actions column, click the Approve/Reject icon next to the name of the service of interest.

    The Approve Request window appears, displaying the details of requested services, including their statuses.

    The Item Status drop-down list is disabled; you enable it and use it in the next steps.

5. Click the Override (Push-Through) icon in the Actions column.

    The tool tip text for this icon is similar to the following: "Click to override pending action."

6. When prompted, confirm that you want to override a pending action not assigned to you.

    The Item Status drop-down list is enabled, meaning that you can now approve or reject each service in the request.

7. If the request contains multiple services, click the alert status icon and enable the Item Status drop-down list for each service that you want to approve or reject.

The status of each service option matches the status of the service during the approval phase of the request life cycle.

Note: Typically, the Status drop-down list includes several options; the names of these options and the order in which they appear are specified in the requestshared.xml file. To update the values and the order of the options in the Status drop-down list, modify the requestshared.xml file to include the status values you want in the order that you want them to appear. For details about modifying the requestshared.xml file, see the "Customizing" chapter of the *Implementation Guide*.

8. Decide whether to approve or reject the request, and then do all of the following that apply:

   ■ As a best practice, if you do not want to approve one or more services, add a note explaining both why the service is being rejected. If applicable, also explain the changes required for the request to be approved.

   ■ Optionally add other notes or attachments as supporting documents.

9. In the Item Status drop-down list, select the appropriate status to approve or reject each service that you are concerned with, and click OK. The statuses are typically named Approved and Rejected, respectively.

   ■ Rejecting a service rejects the entire request. The request is returned to the user who submitted it. That user decides whether to update the request and resubmit it or accept the rejection as final.

   ■ Once all services in the request are approved, the request moves the request to the next phase of the request life cycle, for example, the second level of approval (if applicable) or the fulfillment stage. In other words, after you have approved a service, your service provider's approval processes may require a second level of approval in which case the second level approver is assigned an approval task.

If the approval task was assigned to more than one user or a group, once any assigned user completes that pending action, the request is removed from the Pending Action list for all assigned users. If the request contains multiple services, it is not removed until either all services are approved or one service is rejected.

You are returned to the Request Details window. If you return to the Pending Action list, you can see that the request is removed from your list.

Note: The Approve Request window displays similar information to that shown in the window.

Fulfill Your Own Requests Pending Action

Fulfilling requests pending action is the final essential step in the process of managing requests from creation through fulfillment. All CA Service Catalog users can fulfill requests pending action assigned to them; that is, requests in their own Requests Pending Action queue. However, only administrators and users with the required access control settings can fulfill other users' requests pending action (see page 614).

To fulfill your own request pending action

1. Click Home, Requests.

    Your recent requests are listed, along with links to perform other actions; for example, to view requests pending your action.

2. Click Pending My Action.

    The Requests Pending Action window appears, listing the requests in your queue to be approved, rejected, fulfilled, and so forth.

3. Locate the request and click the Fulfill icon in the Actions column.

    The Fulfill Request window appears.

    Note: The Fulfill Request window displays similar information to that shown in the Request Details window (see page 593).

4. View the details of requested services, including their statuses.

5. Focus on the services for which you are assigned a fulfillment task.

    The status of each service option matches the status of the service during the fulfillment phase of the request life cycle.

    Note: Typically, the Status drop-down list includes several options; the names of these options and the order in which they appear are specified in the requestshared.xml file. To update the values and the order of the options in the Status drop-down list, modify the requestshared.xml file to include the status values you want in the order that you want them to appear. For details about modifying the requestshared.xml file, see the "Customizing" chapter of the *Implementation Guide*.

6. Decide and select the next status for each service for which you are assigned a fulfillment task. Possible statuses include Check Availability, Pending Procurement, Ordered, Shipped, Being Staged, Fulfilled, and Fulfillment Cancelled. The exact statuses available may vary, based on the business processes of your service provider. Two significant statuses are Fulfilled and Fulfillment Cancelled, both of which end the fulfillment process for that service option.

    ■   If you are using CA APM and the service is eligible for asset assignment, select the available inventory of assets and assign one or more assets or indicate no assets are available for assignment. Using the assign assets functionality changes the status of the related service option.

■ If you are *not* using CA APM or the service is *not* eligible for asset assignment, in the Item Status drop-down list, select the appropriate status for each service that you are concerned with.

7. Optionally add notes or attachments as supporting documents.

8. When you are finished, click OK.

You are returned to the Request Details window. If you return to the Pending Action list after all pending action assigned to you have been completed, the request is removed from your list. If the fulfillment task was assigned to more than one user or a group, once any assigned user completes that pending action, the request is removed from the Pending Action list for all assigned users. If the request contains multiple services, it is not removed until either all services are either fulfilled or have their fulfillment cancelled.

Note: After you have indicated your fulfillment actions by changing the status of the services for which you were assigned a fulfillment task, your service provider's fulfillment processes may require additional fulfillment steps; in such cases, the next fulfiller is assigned a fulfillment task.

Fulfill Other Users' Requests Pending Action

Fulfilling requests pending action is the final essential step in the process of managing requests from creation through fulfillment. All CA Service Catalog users can fulfill their own requests pending action (see page 612); that is, requests in their own Requests Pending Action queue. However, only CA Service Catalog administrators and other users with the required access control setting can fulfill requests pending action assigned to other users. The following note explains the required access control setting.

Note: The access control option named Access Control: Proxy Action under Service Builder, Configuration determines which roles have the ability to approve, reject, fulfill, or transfer requests pending action assigned to other users. For details, see the Service Builder configuration settings in the *Implementation Guide*.

To fulfill other users' request pending action

1. Click Home, Requests.

   Your recent requests are listed, along with links to perform other actions; for example, to view requests pending your action.

2. In the Request Lookup section, click Advanced Search.

   The Advanced Search window appears.

3. In the Query drop-down list, do the following:

   a. Ensure that Action Requests is selected.

   b. Optionally, in the Userid field, enter the user ID of interest.

   c. Click Search.

   The Search Results window appears, displaying all requests pending action.

4. Click the name of the request of interest to open it.

   The Requests Details window appears, displaying the details of the request you just opened.

5. View the details of requested services, including their statuses, focusing on the services whose fulfillment status you want to update.

   The status of each service option matches the status of the service during the fulfillment phase of the request life cycle.

   Note: Typically, the Status drop-down list includes several options; the names of these options and the order in which they appear are specified in the requestshared.xml file. To update the values and the order of the options in the Status drop-down list, modify the requestshared.xml file to include the status values you want in the order that you want them to appear. For details about modifying the requestshared.xml file, see the "Customizing" chapter of the *Implementation Guide*.

6. Click the alert status icon in the Actions column.

The tool tip text for this icon is similar to the following: "Click to override pending action."

7. When prompted, confirm that you want to override a pending action not assigned to you.

    The Item Status drop-down list is enabled, meaning that you can now update the fulfillment status of the service.

8. If the request contains multiple services, click the alert status icon and enable the Item Status drop-down list for each service whose fulfillment status you want to update.

    Note: The status of each service option matches the status of the service during the fulfillment phase of the request life cycle.

9. Decide and select the next status for each service whose fulfillment status you want to update. Possible statuses include Check Availability, Pending Procurement, Ordered, Shipped, Being Staged, Fulfilled, and Fulfillment Cancelled. The exact statuses available may vary, based on the business processes of your service provider. Two significant statuses are Fulfilled and Fulfillment Cancelled, both of which end the fulfillment process for that service option.

    ■   If you are using CA APM and the service is eligible for asset assignment, select the available inventory of assets and assign one or more assets or indicate no assets are available for assignment. Using the assign assets functionality changes the status of the related service option.

    ■   If you are *not* using CA APM or the service is *not* eligible for asset assignment, in the Item Status drop-down list, select the appropriate status for each service that you are concerned with.

10. Optionally add notes or attachments as supporting documents.

11. When you are finished, click OK.

    You are returned to the Request Details window. If you return to the Pending Action list after all pending action assigned to you have been completed, the request is removed from your list. If the fulfillment task was assigned to more than one user or a group, once any assigned user completes that pending action, the request is removed from the Pending Action list for all assigned users. If the request contains multiple services, it is not removed until either all services are either fulfilled or have their fulfillment cancelled.

    Note: After you have indicated your fulfillment actions by changing the status of the services for which you were assigned a fulfillment task, your service provider's fulfillment processes may require additional fulfillment steps; in such cases, the next fulfiller is assigned a fulfillment task.

Transfer Your Own Requests Pending Action

You may need to transfer a request pending action assigned to you or other users when you or other users become sick, use vacation time, or otherwise become unavailable to perform or approve an assigned task.

All users can transfer their own requests pending action. However, only CA Service Catalog administrators or other CA Service Catalog users with the required access control setting can transfer other users' requests pending action (see page 618).

For detailed background information and rules about transferring requests and performing related tasks, see Handle Requests Pending Action (see page 604) and Rules for Requests Pending Action (see page 606).

To transfer your own requests pending action

1. Click Home, Requests.

   Your recent requests are listed, along with links to perform other actions, such as view requests pending your action.

2. Click Pending My Action.

   The Requests Pending Action window appears, listing the requests in your queue to be approved, rejected, fulfilled, and so forth.

3. Click the name of the request of interest to open it.

   The Requests Details window appears, displaying the details of the request you just opened.

   The details for the selected request appear, with a Requested Services section at the top of the screen. That section lists all services in the request. If the request contains multiple services, you can transfer them either individually or in groups of two or more at a time.

   Note: A service can be transferred if you can select it. If you cannot select the service, it cannot be transferred. For example, a multi-service request may contain one or more services that are not pending approval or fulfillment and therefore cannot be transferred.

4. Select the service or services and click Transfer.

   The Search User window appears.

   Note: The Search User scope is based on the administration configuration of either all users (global) or only users in the same business unit. For details about setting this configuration, see the sections describing the Administration, Configuration, Options menu in the "Configuring" chapter of the *Implementation Guide*.

5. Review the list of users and select *one* user to whom the selected services are transferred.

You return to the Requested Services section. In the Item Status column, the selected request or requests are marked for Transfer to *username,* where *username* is the user to whom you transferred those requests.

6. Verify that the correct user is designated for the transfer and click OK.

The request pending action is transferred to the user you specified.

Transfer Other Users' Requests Pending Action

You may need to transfer a request pending action assigned to you or other users when you or other users become sick, use vacation time, or otherwise become unavailable to perform or approve an assigned task.

All users can transfer their own requests pending action (see page 616). However, only CA Service Catalog administrators and other users with the required access control setting can transfer requests pending action assigned to other users. The following note explains the required access control setting.

Note: The access control option named Access Control: Proxy Action under Service Builder, Configuration determines which roles have the ability to approve, reject, fulfill, or transfer requests pending action assigned to other users. In addition, the Access Control: Transfer Request Pending Action setting determines the roles that have access to the Transfer button for transferring requests pending action assigned to other users; this button is hidden from other roles. For details about setting both options, see the Service Builder configuration settings in the *Implementation Guide*.

For detailed background information and rules about transferring requests and performing related tasks, see Handle Requests Pending Action (see page 604) and Rules for Requests Pending Action (see page 606).

To transfer other users' requests pending action

1. Click Home, Requests.

   Your recent requests are listed, along with links to perform other actions, such as view requests pending your action.

2. In the Request Lookup section, click Advanced Search.

   The Advanced Search window appears.

3. In the Query drop-down list, do the following:

   a. Ensure that Action Requests is selected.

   b. Optionally, in the Userid field, enter the user ID of interest.

   c. Click Search.

   The Search Results window appears, displaying all requests pending action.

4. Click the name of the request of interest to open it.

   The Requests Details window appears, displaying the details of the request you just opened.

   The details for the selected request appear, with a Requested Services section at the top of the screen. That section lists all services in the request. If the request contains multiple services, you can transfer them either individually or in groups of two or more at a time.

Note: A service can be transferred if you can select it. If you cannot select the service, it cannot be transferred. For example, a multi-service request may contain one or more services that are not pending approval or fulfillment and therefore cannot be transferred.

5. Click the Transfer button at the top of the window.

   The Transfer Request window appears, displaying the details of requested services, including their statuses.

   The Item Status drop-down list is disabled; you enable it and use it in the next steps.

6. Click the alert status icon in the Actions column.

   The tool tip text for this icon is similar to the following: "Click to override pending action."

7. When prompted, confirm that you want to override a pending action not assigned to you.

   The Item Status drop-down list is enabled, meaning that you can now approve or reject each service in the request.

8. If the request contains multiple services, click the alert status icon and enable the Item Status drop-down list for each service that you want to approve or reject.

   The status of each service option matches the status of the service during the approval phase of the request life cycle.

9. Select the service or services and click Transfer.

   The Search User window appears.

   Note: The Search User scope is based on the administration configuration of either all users (global) or only users in the same business unit. For details about setting this configuration, see the sections describing the Administration, Configuration, Options menu in the "Configuring" chapter of the *Implementation Guide*.

10. Review the list of users and select *one* user to whom the selected services are transferred.

    You return to the Requested Services section. In the Item Status column, the selected request or requests are marked for Transfer to *username,* where *username* is the user to whom you transferred those requests.

11. Verify that the correct user is designated for the transfer and click OK.

    The request pending action is transferred to the user you specified.

Delegate a Request Pending Action

You may need to delegate a request when you or other users become sick, use vacation time, or otherwise become unavailable to perform or approve an assigned task. A request pending action can be delegated only once. It can be delegated only by the owner of the request pending action, if the owner is a user who has the required access control setting. The following note explains the required access control setting.

Note: The access control option named Access Control: Delegate Request Pending Action determines which roles have the ability to delegate their own requests pending action. For details about setting this option, see the Service Builder configuration settings in the *Implementation Guide*.

For detailed background information and rules about delegating requests and performing related tasks, see Handle Requests Pending Action (see page 604) and Rules for Requests Pending Action (see page 606).

To delegate a request

1. Click Home, Requests.

   Your recent requests are listed, along with links to perform other actions, such as view requests pending your action.

2. Perform the actions in *one* of the following bullets:

   ■ Click Pending My Action.

      The Requests Pending Action window appears.

      Click the Delegate button for a request pending approval or fulfillment.

   ■ In the Request Lookup section, click Advanced Search.

      The Advanced Search window appears.

      In the Query drop-down list, ensure that Action Requests is selected.

      In the Userid field, enter your user ID and click Search.

      The Search Results window appears, displaying all requests assigned to you.

      Click the Delegate button for a request pending approval or fulfillment.

   The details for the selected request appear, with a Requested Services section at the top of the screen. That section lists all services and service options in the request. If the request contains multiple services and service options, they can be delegated either individually or in groups of two or more at a time.

   Note: A service can be delegated if you can select it. If you cannot select the service, it cannot be delegated. For example, a multi-service request may contain one or more services or service options that are not pending approval or fulfillment and therefore cannot be delegated.

3. Select the service or service options and click Delegate.

   The Search User Accounts window appears.

   Note: The Search User scope is based on the administration configuration of either all users (global) or only users in the same business unit. For details about this configuration, see the sections describing the Administration, Configuration, Options menu in the "Configuring" chapter of the *Implementation Guide*.

4. Review the list of users and select *one* user to whom you delegate the selected services or service options.

   You return to the Requested Services section. In the Item Status column, the selected service or service options are marked for delegation to *username,* where *username* is the user whom you selected.

5. Verify that the correct user is named for the delegation and click OK.

   The request pending action is delegated to the user you specified.

Auto-Delegate Your Own Requests Pending Action

You can optionally use auto-delegation to specify that all of your newly assigned or transferred requests pending action are automatically delegated to another user. Auto-delegation is typically useful for managers who monitor requests but who do not address requests directly. Auto-delegation is also useful for users who are out of the office for travel, vacation, or another reason. Thus, auto-delegation helps ensure that requests are addressed in a timely manner.

For auto delegation to function correctly for a requested service, the process definition that governs the service must assign the request to a group before assigning the request to an individual user. Otherwise, requests pending action for the service may be incorrectly auto-delegated to a user in spite of being assigned to a group, which may cause an error in the approval or fulfillment process.

Note: The access control option named Display Auto-Delegation on User Profile under Administration, Configuration determines whether users have the ability to view and change auto-delegation settings, according to user roles that have access to User Management UIs. For details, see the administration configuration settings in the *Implementation Guide*.

To auto-delegate your own requests pending action

1. Move the cursor to the top right portion on the screen and click the Profile button next to the Help button.

   The User Profile window appears; the fields are read-only.

2. Click the Edit button.

   The fields on the window open for editing, and the title of the window changes to Edit User Profile.

3. Click the down-arrow icon next to the Request Auto-Delegation heading.

   The Request Auto-Delegation setting for your user ID opens, so that you can view and optionally change it.

4. To remove an existing auto-delegation (if applicable), click the red "minus sign" icon.

5. To add or change auto-delegation, click the magnifying glass icon.

   The Search and Select User Search dialog appears.

6. Review the list of users.

   Note: The Search User scope is based on the administration configuration of either all users (global) or only users in the same business unit. For details about setting this configuration, see the sections describing the Administration, Configuration, Options menu in the "Configuring" chapter of the *Implementation Guide*.

7. Click the name of the user to designate for auto-delegation.

   The dialog closes, and you return to the Edit User Profile window. The user you selected is named in the Delegate field.

8. Verify that the correct user is designated for auto-delegation and click OK.

9. Click Done.

   You return to the original window that was open before you clicked the Profile button.

All new requests pending action for your user ID is auto-delegated to the user named in the Delegate field.

Auto-Delegate Other Users' Requests Pending Action

You can optionally use auto-delegation to specify that any *new* requests assigned to a specific user are automatically delegated to a different user instead. Here, *new* request means any newly assigned or transferred pending action. Auto-delegation is useful for managers who monitor requests but who do not address requests directly. Auto-delegation is also useful for users who are out of the office for travel, vacation, or another reason. Thus, auto-delegation helps ensure that requests are addressed in a timely manner.

For auto delegation to function correctly for a requested service, the process definition that governs the service must assign the request to a group before assigning the request to an individual user. Otherwise, requests pending action for the service may be incorrectly auto-delegated to a user in spite of being assigned to a group, which may cause an error in the approval or fulfillment process.

Note: The access control option named Display Auto-Delegation on User Management under Administration, Configuration determines whether users have the ability to view and change auto-delegation settings, according to user roles that have access to User Management UIs. For details, see the administration configuration settings in the *Implementation Guide*.

To auto-delegate other users' requests pending action

1. Click Administration, Users.

   The Users window appears.

2. In the User ID field, enter search criteria to find the user whose requests you want to auto-delegate to another user.

   The Users search results appear.

3. Click the name of the user whose requests you want to auto-delegate.

   The User Profile window appears, displaying configuration details for the user you selected.

4. Click Edit.

   The fields on the window open for editing, and the title of the window changes to Edit User Profile.

5. Click the drop-down icon next to the Request Auto-Delegation heading.

   The Request Auto-Delegation setting opens, so that you can view and optionally edit it.

   The user whose name appears in the Delegate field, if any, is the one to whom requests are currently auto-delegated.

6. To remove an existing auto-delegation (if applicable), click the red "minus sign" icon.

7. To add or change auto-delegation, click the magnifying glass icon.

   The Search and Select User Search dialog appears.

8. Review the list of users.

   Note: The Search User scope is based on the administration configuration of either all users (global) or only users in the same business unit. For details about setting this configuration, see the sections describing the Administration, Configuration, Options menu in the "Configuring" chapter of the *Implementation Guide*.

9. Click the name of the user to designate for auto-delegation.

   The dialog closes, and you return to the Edit User Profile window. The user you selected is named in the Delegate field.

10. Verify that the correct user is designated for auto-delegation and click OK.

    You return to the Edit User Profile window.

11. Click Done.

    You return to the Administration, Users window.

All new requests for the user whose profile was edited are auto-delegated to the user named in the Delegate field.

Take or Return a Request Pending Action

From the Take/Return window, you can take or return ownership of requests pending action. You can take ownership of requests pending action from either a group queue or the queue of another user in the same business unit or sub-business unit. Similarly, you can return ownership of requests pending action from your queue to your group's queue. You may need to take or return ownership of requests pending action assigned to your group.

The Take and Return functions work *only* if you are logged in as a user of the group to which the request pending action is assigned. In addition, only CA Service Catalog administrators and other users with the required access control setting can take or return requests for a group to which they belong. The following note explains the required access control setting.

Note: The access control option named Access Control: Take/Return Request Pending Action under Service Builder, Configuration determines which roles have the ability to take or return requests pending action for a group to which they belong. For details, see the Service Builder configuration settings in the *Implementation Guide*.

To take or return a request pending action

1. Click Home, Requests.

   Your recent requests are listed, along with links to perform other actions, such as view requests pending your action.

2. Click Pending My Action.

   The Requests Pending Action window appears.

   Click the Take/Return button for a request pending approval or fulfillment.

   The Take/Return window appears, showing the details for the selected request, including a Requested Services section at the top of the screen. That section lists all services and service options in the request. If the request contains multiple services or service options, they can be taken or returned either individually or in groups of two or more at a time.

   Note: A service or service option can be taken or returned if you can select it. If you cannot select the service or service option, it cannot be taken or returned. For example, a multi-service request may contain one or more services or service options that are not pending approval or fulfillment and therefore cannot be taken or returned.

3. To *take* one or more services from a request pending action, select the services or service options and click Take.

   The request is marked to be taken by your user ID.

If a request is already taken, the status indicating "taken by *username*" appears next to the Item Status field of the request item, and therefore you cannot take it. The *username* identifies the user who has taken the request. Other users in the same group can still approve or fulfill a taken service, if necessary. As a prerequisite, however, either the user who has taken the service must return it or an authorized user must transfer it to the new assigned user. An authorized user is a CA Service Catalog administrator or has the required access control setting, as noted earlier in this topic.

If a request is not taken but the Take option is not available, then the action is not assigned to your group.

4. To *return* one or more service or service options from a request pending action, select the services or service options and click Return.

   The only services that you can return are those already marked as taken by your user ID; these are requests that you have previously taken.

5. Click OK to perform the take or return action.

   Your action is confirmed: Any requests that you have taken are marked as taken by your user ID. Similarly, any requests that you have returned are no longer marked as taken, meaning that they are available for other users in your group to take.

Optionally, you can perform another action on the Take/Return window or access another window.

Ignore, Retry, or Override Alerts

At any point during the request life cycle, including the submitted state or during the approval process, requests may occasionally become marked with an alert status indicating a potential problem, typically due to system error or user error. The alert is indicated by yellow warning icon in the Status column of several request windows, including the Open Requests, Completed Requests, Pending My Action, and Request Search windows. This icon also appears on the Recent Request Lists, if applicable. If the role of a user does not permit access to this option, then that user cannot see the warning icon, even if the user's request is stuck.

As a best practice, it is recommended that you ignore, retry, or override (push through) alerts as soon as you become aware of them.

Only CA Service Catalog administrators and other users with the required access control setting can ignore, retry, or override alerts. The following note explains the required access control setting.

Note: The access control option named Access Control: Push Through Request under Service Builder, Configuration determines which roles have the ability to ignore, retry, or override alerts. In addition, the access control option named Access Control: Display Request Warning determines which roles are able to see the warning icon indicating that a request is stuck. For details about setting these options, see the Service Builder configuration settings in the *Implementation Guide*.

For detailed background information and rules about processing requests marked with an alert status and performing related tasks, see <u>Handle Requests Pending Action</u> (see page 604) and <u>Rules for Requests Pending Action</u> (see page 606).

To ignore, retry, or override alerts

1. Click Home, Requests.

2. Find the "alert" request on any of the following windows: Open Requests, Completed Requests, Pending My Action, Request Search, or Recent Requests Lists.

   The request may be marked with an alert status, because of one or more information-only alerts logged during the request flow. In this case, ignore the alerts, as follows:

   a. Open the window that includes the alert request.

   b. Click the Push Through option next to the request with the alert status icon.

   c. Open the Retry Failed Actions window and click Ignore Checked Alerts.

   The alerts are made inactive, and the warning icon is removed.

3. Click the Push Through link for the alert request.

4. The Retry Failed Actions window appears.

   Review the alerts listed on that page and decide an action, as follows:

a. For each request, review the text provided by CA Service Catalog about why it was marked as an alert, fix the problem, and retry the intended action.

For example, if a CA Workflow shutdown, CA IT PAM shutdown, or other system-related error may have caused the problem, restart CA Workflow, CA IT PAM, or another failed component and retry the action.

Thus, if the request became stuck when a user attempted to submit it, and if the retry action was successful, then the request moves to the next valid state (Pending Approval). The approver can now approve the request.

b. If the request remains marked as an alert even after retries, consider overriding the alert (pushing through the request to the next status), as noted in the next step.

Note: When the status of a requested service is either Pending or Completed, you can either ignore or push through the alert, using your best judgment as an administrator.

5. If necessary, click the Push Through Request tab.

6. The Push Through Request window appears.

The details for the selected request appear, with a Requested Services section at the top of the screen. That section lists all services in the request. If the request contains multiple services, they can be pushed through either individually or in groups of two or more at a time.

7. For each stuck service, click the Item Status drop-down list and select the Approval or Fulfillment status that best meets your needs.

The status options for any service vary, according to its current status. Use the drop-down list to view all available options.

8. Click OK.

Each service whose status you changed has its alerts removed (overridden) and is "pushed through" to the selected status.

Delegation of Catalogs

Delegating the use of your catalog to another user (a delegate) enables the delegate to browse your catalog and to create and submit requests on behalf of you (the delegator). A delegator's catalog typically contains services that are not available in the delegate's catalog.

Delegators are typically executives or managers but can be any user whose role permits creating and submitting requests. Delegates are typically executive assistants or other employees who report to the delegator but likewise can be any user whose role permits creating and submitting requests.

Typically, a delegate creates and submits requests for the delegator from the delegator's catalog. While a delegate is creating a request from a delegator's catalog, the Requested For field shows the delegator's name, and the field cannot be changed.

When applicable, all history, auditing, and logging records for a request show that the delegate created and submitted the request on behalf of the delegator.

Note: As noted earlier, a user can be a delegator or delegate if the role of the user permits creating and submitting requests. Administrators assign this right to individual roles by setting the configuration option named Access Control: Add Request in the Request Management Configuration section of the Administration Configuration page. For details about this parameter and related ones, see the *Implementation Guide*.

How the Delegation of Catalogs Works

Delegating the use of your catalog to another user enables that user to browse your catalog and to create and submit requests containing services that may not be available in the user's own catalog. Delegates typically create and submit such requests for you (the delegator) from your catalog.

1. Administrators review the basic information about delegating catalogs (see page 629). Also review the following:

   ■ Comparison to related features (see page 630)

   ■ Limitations of delegation of catalogs (see page 631)

   ■ Considerations for business units (see page 632)

   ■ Sample scenarios (see page 633)

2. Administrators configure CA Service Catalog to enable delegation of catalogs (see page 634).

3. Users (delegators) edit their profile to delegate the use of their catalogs (see page 635) to one or more other users (delegates).

4. The delegates use the catalog of a delegator (see page 637) to create, edit, and submit requests.

Comparison to Related Features

Delegation of catalogs is related to, but not the same as, the following features:

■ Requesting a service for another user, using your own catalog, by specifying the other user's name in the Requested For field when you view and change additional information (see page 570) in the request before submitting it.

■ Delegating a request (see page 620) means that you manually delegate a request pending action (a submitted request awaiting approval), from either yourself or another request manager, to a different request manager. The delegated request manager can then approve, reject, or transfer the request.

■ Auto-delegating your own requests pending action (see page 621) or auto-delegating other users' requests pending action (see page 623)   means that you automatically delegate such requests from your own requests pending action queue or another request manager's queue to a different request manager's queue, typically on a temporary basis. The auto-delegated request manager can then approve, reject, or transfer the request.

In contrast to these features, delegating the use of your catalog enables another user to browse *your* catalog, to create, edit, and submit requests on your behalf. Typically, your catalog contains services that are not available in your delegate's catalog.

Limitations of Delegation of Catalogs

The following limitations apply when you delegate the use of your catalog to another user:

■ If you have created requests but have not submitted them, your delegates *cannot* view, update, or submit those requests on your behalf. Delegates must create and submit requests themselves from the delegator's catalog.

■ While using your catalog, delegates *cannot* change the Requested For field to request services for any user other than you (the delegator).

■ Users who have a role that cannot create a request (such as Service Manager) *can* delegate their catalogs. However, delegates of such users *cannot* browse or create requests from such delegators' catalogs. Therefore, as a best practice, users whose roles prohibit them from creating requests do *not* delegate their catalogs.

■ All users can delegate the use of *their own catalog* to another user. However, if you have the Super Business Unit Administrator (stadministrator) or Service Delivery Administrator (spadministrator) role, you can delegate the use of either *your own catalog* or *the catalog of any other user ID* to another user. If an administrator delegates the use of your catalog to another user, neither you nor the other user is notified automatically by the Catalog system. However, in such cases, the delegate does appear in the list of delegates on your User Profile page, and you can manually remove that delegate if necessary. As a best practice, administrators who perform such actions notify affected users personally; for example, using email.

Important! The delegate must have a role in the business unit of the delegator. Otherwise, the delegator's catalog is not made available to the delegate.

Considerations for Business Units

Any user can delegate the use of his or her catalog to any other user in any business unit. You can delegate your catalog to a user in your own business unit or another business unit, including the top-level (service provider) business unit.

If you delegate your catalog to a user in the same business unit, the delegate can use your catalog immediately. In other words, when the delegate logs in to CA Service Catalog, the delegate can see and select your name in the Use Catalog Of field of the Requests home page. (This page appears when you select Home, Requests) After doing so, the delegate browses and uses your catalog.

However, when you delegate your catalog to a user in another business unit, you must have a role in that business unit. Otherwise, the delegate cannot see and select your name in the Use Catalog Of field of the Requests home page and thus cannot use your catalog. Therefore, whenever you delegate your catalog to a user in another business unit, verify that you have a role in that business unit. If necessary, you or an administrator must edit your user profile (see page 86) to add a role for your user ID in that business unit.

For example, suppose your organization is grouped into business units according to department. If you work in the Finance department and you delegate use of your catalog to a user in the Human Resources department (a different business unit), verify that you have a role in Human Resources to make the delegation meaningful.

Sample Scenarios

The following sample scenarios illustrate possible instances of delegating the use of a catalog--either your own catalog or another user's catalog.

Scenario 1

A very busy CIO leaving for an intensive conference delegates her catalog to her executive assistant, with instructions to create and submit requests for ten items on her behalf.

The items are not available in the assistant's catalog, so the CIO delegates use of her catalog to the assistant, to help ensure that the requests are submitted in time. During the CIO's absence, the assistant uses the CIO's catalog as a delegate and requests the ten items for the CIO.

All history, auditing, and logging records for the requests show that the assistant created and submitted the requests on behalf of the CIO.

Scenario 2

This scenario concerns an administrator (User A) delegating the use of the catalog of another user (User B) to a different user (User C). Doing so requires the Super Business Unit Administrator (stadministrator) or Service Delivery Administrator (spadministrator) role.

In this scenario, the same CIO must leave for the same conference but requires the ten items requested not for herself but for a vice president who reports to her. That vice president is on leave without access to CA Service Catalog but needs the items when returning to work next week.

In this scenario, the CIO delegates the use of the vice president's catalog (not her own catalog) to her assistant. During the CIO's absence and before the vice president returns to work, the assistant uses the vice president's catalog as a delegate and requests the ten items for the vice president.

All history, auditing, and logging records for the requests show that the assistant created and submitted the requests on behalf of the vice president.

Scenario 3

The same CIO begins to create a request for herself but must stop suddenly and save the request to address an emergency issue. When she realizes that the issue requires long, uninterrupted effort, she stops the effort, cancels the request, and instructs her assistant to create and submit the request for her as a delegate.

When applicable, all history, auditing, and logging records for the request show that the delegator created the request, but the assistant edited and submitted the request on behalf of the delegator.

Enable or Disable Delegation of Catalogs

Users with the Super Business Unit Administrator (stadministrator) or Service Delivery Administrator (spadministrator) role can enable or disable delegation of catalogs within the business unit--and any sub-business units--to which their role applies. Users with other roles can optionally delegate the use of their catalogs to any other user. Delegation of catalogs is enabled by default, but you may need to enable it explicitly if it was disabled earlier.

To enable or disable delegation of catalogs

1. Click Service Builder, Configuration, Request Management Configuration.

    The Service Builder configuration page appears, with the Request Management Configuration section in focus.

2. Locate the option named Enable Delegation of Catalog: set this option to Yes to enable the feature or No to disable it.

    -   If this option is set to No, you can optionally enable it by clicking the Edit (pencil) icon for the option, checking the box, and clicking Update Configuration.

    -   If this option is set to Yes, you can optionally disable it by clicking the Edit (pencil) icon for the option, removing the check from the box, and clicking Update Configuration.

When this option is enabled, no users are *required to* delegate use of their catalogs, but they can *optionally* do so.

Note: The delegation of a catalog does *not* affect the Catalog configuration of the Service Builder, such as the Use Service Provider Catalog Only and Pass Through Catalog configuration parameters. Similarly, the delegation of a catalog is *not* directly related to the configuration options named Access Control: Add Request, Access Control: Edit Request, and so forth that also exist in the same Request Management Configuration section as Enable Delegation of Catalog. For details about any of these other parameters, see the *Implementation Guide*.

Delegate the Use of a Catalog

When you delegate the use of your catalog to other users, they can browse your catalog, and they can create and submit requests on your behalf. Typically, you delegate the use of your catalog to other users when you are unavailable for some length of time to perform this work yourself but the work needs to continue in a timely manner with minimal supervision, to help keep business operations flowing smoothly. As a prerequisite, review the limitations of delegation of catalogs (see page 631).

To delegate the use of a catalog

1. Verify with your Super Business Unit Administrator (stadministrator) or Service Delivery Administrator (spadministrator) that this feature is enabled.

2. Click Home, Administration, Users.

3. Click the Search button on the Users search dialog.

    The user IDs that you can update appear.

4. Click the Edit (pencil) icon for the user ID you want to edit.

    The Edit User Profile page for the selected user appears.

5. Click the Open (triangle) icon for the Delegate Use of Catalog box.

    The box opens so that you can update it.

6. Click the Search (magnifying glass) icon next to the Delegates field.

    The list of eligible user IDs appears.

7. Click the user ID to which you want to delegate use of your catalog.

    The user ID you selected is added to the Delegates field.

8. Click the Add (plus sign) icon.

    The user ID you selected is moved from the Delegates field to the list of delegates underneath it. The user ID is thereby added to your list of delegates.

    Important! The delegate *must* have a role in the business unit of the delegator; otherwise, the delegator's catalog is *not* made available to the delegate!

    Also keep in mind the following:

    ■ Delegates *cannot* edit and submit existing requests created by the delegator.

    ■ A single user can be a delegate for multiple delegators.

    ■ Delegates can create and submit requests using the delegator's catalog.

9. If applicable, add another delegate by clicking the Search icon and repeating the previous two steps.

10. When you are finished adding delegates, click OK.

If you delegated the use of your own catalog, all delegates you selected can browse your catalog and can create, edit, and submit requests on your behalf. If you are a Super Business Unit Administrator or Service Delivery Administrator who delegated the use of the catalog of another user, the delegates you selected can browse the catalog and can create, edit, and submit requests on behalf of that user.

Remove a Delegate

When you remove a user from your list of delegates, the user can no longer browse your catalog or create and submit requests on your behalf. Remove a user from your list of delegates when appropriate for business reasons; for example, if the delegate leaves your organization or your department, if you transfer to a different department, if you no longer need the user as a delegate, and so forth. Typically, you remove delegates from using your own catalog. However, if you are a Super Business Unit Administrator or Service Delivery Administrator you can remove delegates from using the catalog of another user.

To remove a delegate

1. Click Home, Administration, Users.

2. Click the Search button on the Users search dialog.

    The user IDs that you can update appear.

3. Click the Edit (pencil) icon for the user ID you want to edit.

    The Edit User Profile page for the selected user appears.

4. Click the Open (triangle) icon for the Delegate Use of Catalog box to reveal the user's list of delegates.

    The box opens so that you can update it.

5. In the list of delegates, select the user or users whom you want to remove as delegates and click the Delete (minus sign) icon.

    The user or users you selected are removed from the list of delegates.

6. When you are finished removing delegates, click OK.

If you removed delegates from using your own catalog, all delegates you removed can no longer browse your catalog and can no longer create, edit, and submit requests on your behalf. If you are a Super Business Unit Administrator or Service Delivery Administrator who removed delegates from using the catalog of another user, the delegates you removed can no longer browse the catalog and can no longer create, edit, and submit requests on behalf of that user.

Use the Catalog of a Delegator

When another user, perhaps your manager, specifies you as a delegate, you can use the delegator's catalog instead of your own. The delegator's catalog likely includes more services and service options than your own. After accessing the delegator's catalog, you can browse it and use it create, edit, and submit requests on the delegator's behalf.

To use the catalog of a delegator

1. Click Home, Requests.

   The Requests window appears.

2. Click the Use Catalog Of drop-down list.

   The list displays your delegators, the users for whom you are a delegate in your currently logged in business unit (tenant) or one of its sub-business units (sub-tenants).

   If no users appear:

   - Contact users for whom you believe you are a delegate and verify that they have delegated the use of their catalogs (see page 635) to you.

   - Verify the business unit in which the delegator has specified you as a delegate. Verify that you have a role in the business unit.

   - Verify that the delegation of catalogs (see page 634) option is enabled.

3. Select the delegator for whom you want to create, edit, and submit requests.

You can now browse the delegator's catalog and create, edit, and submit requests on the delegator's behalf. When applicable, all history, auditing, and logging records for a request shows that the delegate created, edited, and submitted the request on behalf of the delegator.

Management of Unsubmitted Requests Created by Delegates

A delegate can create and save requests using a delegator's catalog and submit them later, on behalf of the delegator. However, before the delegate submits the request, the following events may occur:

- The delegator (or an administrator) removes a delegate (see page 636) from using the delegator's catalog.

- An administrator disables the delegation of catalogs (see page 634) in one or more business units or throughout the Catalog system

In both cases, delegates are *not* notified automatically of the event, and delegates are no longer able to submit the requests. The only exception is that delegates whose roles permit them to override alerts (see page 627) can optionally "push through" the request to the next status and thereby submit it. Similarly, delegates can optionally delete such requests, because delegates own the requests.

In addition, delegators can view such requests made on their behalf (whether submitted or not) in the Open Requests queue; however, delegators *cannot* delete or submit such requests. The only exception is that delegators whose roles permit them to override alerts can optionally "push through" the request to the next status and thereby submit it.

As a best practice, delegators and delegates decide together how to manage these saved but unsubmitted requests, so that such requests do not remain open long term. Options are as follows:

- Delete the request; Only the delegate can perform this action, as explained earlier

- Push-through the request; The delegate, delegator, or an administrator with the required role (upon request) can perform this action, as explained earlier

- *Either* temporarily re-add the user as a delegate *or* request an administrator to re-enable the delegation of catalogs in your business unit—whichever is applicable—long enough for the delegate to submit the request

Glossary

accounts

Accounts are part of a business unit. Accounts are used to subscribe and request services. Charges are applied at the account level in CA Service Accounting.

action

An *action* is the smallest unit of work in the Rule Management Engine. This task is executed when a rule gets notified. Examples include sending an email, running a script, and running Java code.

Activity Based Costing (ABC)

Activity Based Costing (ABC) is a costing methodology used to trace overhead costs directly to cost objects, that is, products, processes, services, or customers and help managers to make the right decisions regarding product mix and competitive strategies.

Activity Based Management (ABM)

Activity Based Management (ABM) uses Activity Based Cost (ABC) information to allocate resources more effectively during budgeting and planning processes.

adjustments

Adjustments are credits or debits applied to services, individual charges of a Service Option Group, and SLA violations. These adjustments can be made by a fixed dollar amount or a percentage. Several types of general and SLA violation adjustments are available.

batch print job

A *batch print job* is a collection of invoices that have been grouped according to the specified credentials. After the collection has been created, it remains unchanged and its invoices can be printed as a whole or individually at any time.

business unit

The *business unit* is a branch in the organizational structure. This organizational unit has some characteristics of the service provider and all characteristics of a sub-department.

CA Configuration Management Database (CA CMDB)

CA Configuration Management Database (CA CMDB) is a functional data repository that unifies and simplifies the management of configuration information. CA CMDB consolidates and reconciles disparate sources of IT-related data in the context of business priorities. CA CMDB provides visibility into configuration item information such as resource attributes, relationships and dependencies.

CA Embedded Entitlements Manager (CA EEM)

CA Embedded Entitlements Manager (CA EEM) provides enterprises with the ability to manage the identities of their employees, partners, and customers, provision web services and applications to them, and define and enforce authentication and authorization policies. This product is also known as:

- eTrust™ IAM Toolkit

- eTrust Embedded Identity and Access Management (eIAM)

catalog

See service catalog (see page 643).

cost allocation

Cost allocation is a method to determine the cost of services provided to users of that service. It does not determine the price of the service, but rather it determines the cost of providing the service.

cost element

A *cost element* is a resource used to subdivide costs corresponding to the consumption of a particular service.   It is the amount paid for a resource consumed by an activity and included in the activity cost pool. This information helps your organization validate that the total overhead calculated at the beginning of the process matches the total obtained when summing the overhead that is assigned to each individual product using ABC.

cost pool

A *cost pool* is a grouping of all cost elements associated with an activity.

Dashboard Library

A *Dashboard Library* is a hierarchical tree structure of name spaces facilitating the sharing of information. The Dashboard Library does not hold actual data.   Instead, it stores information about how the data can be accessed.

data collector (DC)

A *data collector* collects data for particular metrics.

data mediation profile

The *data mediation profile* is a definition of the data feed and structure of the external data. It also provides powerful features to manipulate and normalize usage event data.

data object

A *data object* defines the data for a chart or table. The source of data for a data object can be a SQL database, comma separated file or any other data source that can be accessed by a Java plug-in. Data objects are managed in folders to help categorize them by purpose.

data view

A *data view* formats the data produced by a data object. Data can be presented in tabular or chart format or both. Data views are managed in folders to help categorize them by purpose.

direct costs

Direct costs are those costs that can be easily traced and assigned to activities.

dynamic invoice group

Dynamic invoice groups are invoice groups that contain account lists that are generated dynamically given specified criteria stored as a data object.

eTrust Embedded Identity and Access Management (eIAM)

See CA Embedded Entitlements Manager (CA EEM) (see page 640).

event filter

An *event filter* can be specified as part of a rule. An event filter is used to further refine the event. This allows actions associated with a rule to be invoked only if certain conditions in an event are met.

failover

Failover is the process of maintaining an up-to-date copy of a database on an alternate system for backup. Traditional failover architecture is comprised of one system running the application while the other is idle in standby mode, ready to take over if the primary system fails. An alternative architecture includes clustering.

invoice group

An *invoice group* is comprised of a set of accounts that can be run together in one instance.

ITIL

ITIL (IT Infrastructure Library) is essentially a series of documents that are used to aid the implementation of a framework for IT Service Management. This customizable framework defines how service management is applied within an organization. Although ITIL was originally created by the CCTA (a UK Government agency), it is now being adopted and used across the world as the de-facto standard for best practice in the provision of IT Service. Although ITIL covers a number of areas, its main focus is on IT Service Management.

Java Runtime Environment (JRE)

Java Runtime Environment provides the minimum requirements for executing a Java application. It consists of the Java Virtual Machine (JVM), core classes, and supporting files. Also known as Java Runtime, JRE is part of the Java Development Kit (JDK), a set of programming tools for developing Java applications.

Management Database (MDB)

The *Management Database (MDB)* is the CA Management Database schema. The MDB is a common enterprise data repository that integrates CA product suites. The MDB provides a unified database schema for the management data stored by all CA products (mainframe and distributed). Use of the MDB with CA products enables full integration for managing your IT infrastructure. The MDB integrates management data from all IT disciplines and CA products. Customers may extend the MDB Schema to include additional IT management data from non-CA software products and tools. For more information about the MDB, see the MDB documentation included with CA Service Catalog.

pool worksheets

Pool worksheets are used to trace cost elements to activities to obtain dollar values for each related activity cost pool. Formulas can be used to adjust costs spanning across fiscal periods and sets.

process definition

A *process definition* is a representation of one of your business processes. A process definition is comprised of nodes, events, roles, actors, work, and the criteria for process logic. Running a process definition creates a process instance. You can create multiple process instances of the same process definition.

process instance

Whereas process definitions represent what you want to happen in your business process, *process instances* represent what is actually happening. By running a process definition, you create a process instance. You can create multiple process instances of the same process definition. Process instances are sometimes called a process definition instance.

proration

Proration is the process of dividing the cost of a subscription over the billing cycle period of an account.

report layout

A *report layout* is used to present multiple report elements as one report. With report layouts, you can design customized reports by using objects such as text, images, URLs, and most importantly, multiple data view objects. You can choose positions, sizes, and colors, borders/styles, set up snap to grid, and many other tools to obtain the desired result. Layouts are managed in folders to help categorize them by purpose.

routing nodes

Routing nodes define the branching of control flow in a Workflow process definition. Choice routing nodes allow one route for the next activity. Parallel routing nodes allow more than one route to be taken.

rule

A *rule* is associated with an event. A rule can have a set of filter conditions that define when the rule is applied. When the filter conditions are satisfied, the rule actions are launched, if the rule is enabled.

service

A *service* is a product, application or something you want to make available to your clients, through a request or a subscription. A service must be defined by associating one or more service option groups to the service before it can be subscribed to or requested.

service catalog

A *service catalog* consists of services published by a business unit or throughout the enterprise. Services are built of one or more service option groups that describe IT services and how to charge for them. The service catalog enables an organization to model its business units or departments and manage the users accounts contained within those units. The service catalog also defines the inclusion, inheritance, dependencies, and associative relationships between each of the published items (services, service option groups, and service option group definitions) in the catalog. Accounts and users subscribe to or request services contained within the catalog. Services in the service catalog may be organized into folders and may contain detailed information about the price of a service. Services may represent one or more metrics and include service level agreements.

Service Level Agreement (SLA)

A *Service Level Agreement* is a contract that specifies the level of service provided while the agreement is in effect.

Service Level Objective (SLO)

A *Service Level Objective* is the building block for an SLA.   When creating an SLA, the SLO provides the business objectives or business rules that set the warning and violation threshold levels; thus the success or failure of the overall SLA.

service option element

A *service option element* is the smallest unit of the catalog that defines a piece of text, fee, application, and so on, in a service. Service option elements are grouped or classified in service option groups.

service option group

A *service option group* contains the cost associated with subscribing to a service, based on transaction, usage based billing, rate and more. It can be composed of many types of chargeable and non-chargeable items, including, for example, rates, applications and agreements.

service worksheets

Service worksheets contain services, service option groups, and service option elements. Service worksheets are used to apply applicable costs to associated services. These costs can be direct or derived from associated activity cost pools. Formulas may be used to adjust costs spanning across fiscal periods and sets.

Simple Object Access Protocol (SOAP)

SOAP is a lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment that uses HTTP as the transport protocol and XML as the payload encoding scheme. Developers can use whatever programming language they are familiar with to call exposed methods of CA Service Catalog, using standard SOAP call syntax.

worksheets

Worksheets enable you to define costs related to business activities. There are two types of worksheets: Service Worksheet and Pool Worksheet.

WSDL

WSDL (Web Services Description Language) is an XML format published for describing web services.